

WOA: Pulsar's Power PC BOARD FOR THE A500

100%
AMIGA

Amazing

COMPUTING™

Your Original AMIGA® Monthly Resource

For The Commodore

AMIGA®

Volume 5 No. 6 June 1990
US \$3.95 Canada \$4.95

**POWER USER
ISSUE**

Programming:

Object-Oriented Programming in C++
Amiga Turtle Graphics
APL: Primitive Functions
Convergence:
Part Five In The Fractals Series

Productivity:

WordPerfect Macros

Reviews:

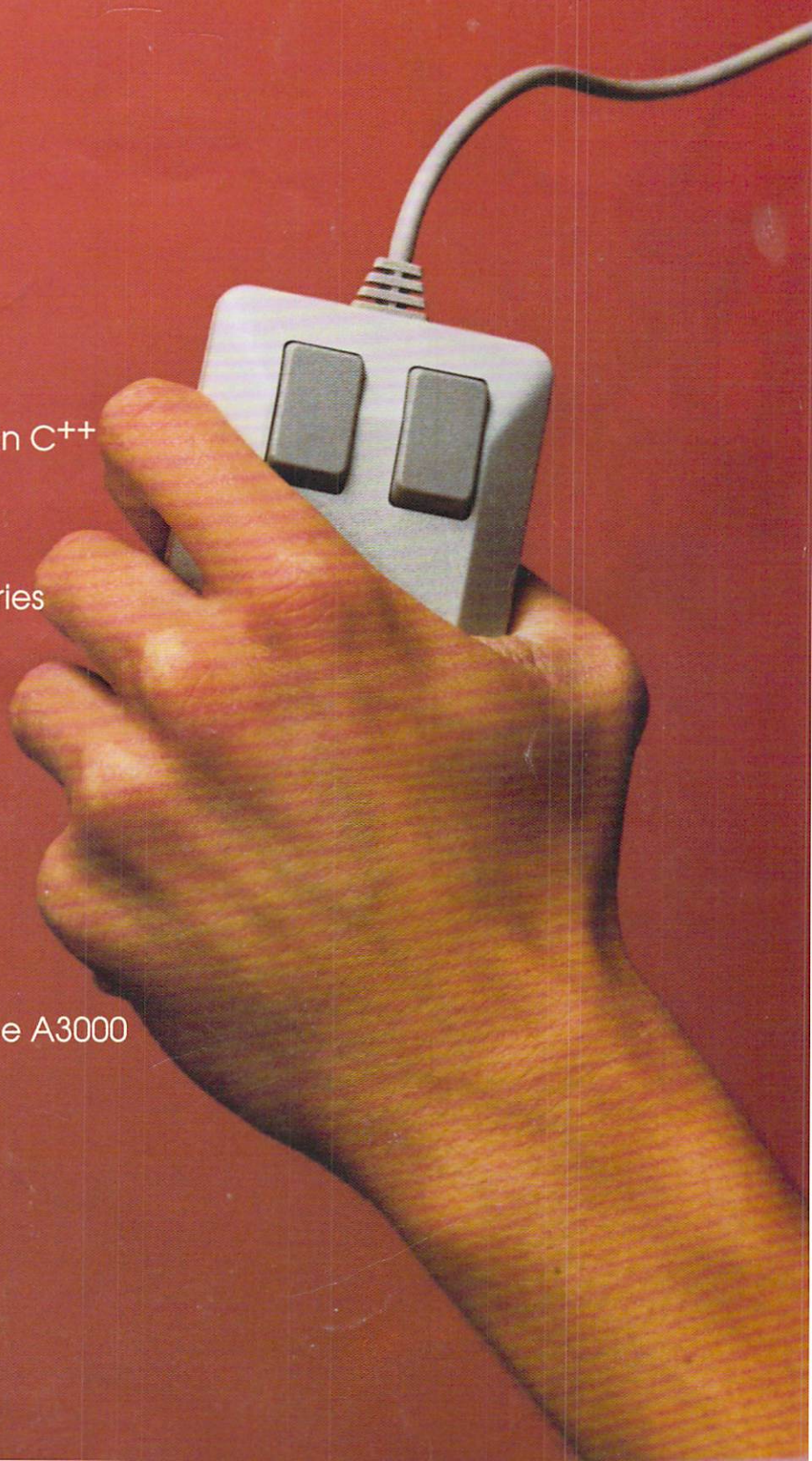
DigiMate III
PageStream 1.8

Hardware Project:

RAM Expansion For Your A500

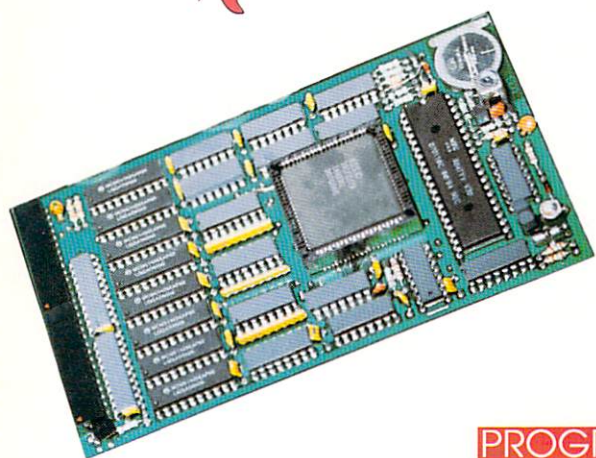
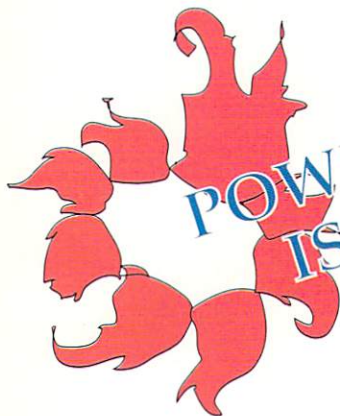
PLUS!

CBM'S Multimedia Launch Of The A3000
Workbench 2.0's Display Modes



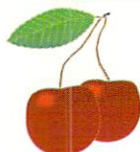
CONTENTS

POWER USER ISSUE



Pulsar's POWER PC BOARD makes your Amiga 500 PC-compatible. Read about this and other newly released Amiga products in our article, *World of Amiga, New York* show coverage, page 47.

INSIDE STORIES



Cherryware 39

by Julie Petersen

Extend your vocabulary with computerisms that define products and describe characteristics associated with their use.

World of Amiga, New York 47

Commodore, over 40 other Amiga vendors, and more than 15,000 Amiga users gather for a special two-day event.

Commodore's Launch 81

CBM defines multimedia as it presents the Amiga 3000.



Workbench 2.0 87

Display à la mode.

PROGRAMMING

Convergence 41

by Paul Castonguay
Part Five of the Fractal series.

C++ 51

by Scott B. Steinman, O.D., Ph.D.
An introduction to object-oriented Amiga programming.

APL and the Amiga 69

by Henry T. Lippert, EdD
Primitive functions and their execution.

Amiga Turtle Graphics 57

by Dylan McNamee
Computer graphics and programming with a LOGO-like graphics system.



HARDWARE

Rapid Fire Joystick 21

by John Iovine
Build a rapid fire joystick circuit for game playing on the Amiga.



The AM 512 35

by James Bentley
Upgrade your A500 to a 1 megabyte machine.



ALL
LISTINGS
ON
DISK

TUTORIALS AND REVIEWS

PageStream 1.8 14

by John Steiner

Putting Soft-Logik's desktop publishing package to work.

WordPerfect Macros 18

by Mike Hubbart
Create time-saving macros in WordPerfect.

Mail Order Macros 20

by Armando Cárdenas
Address envelopes using WordPerfect macros.

DigiMate III 31

by Frank Mc Mahon
Animation comes to DigiPaint.

Pipe Dream 88

by Miguel Mulet
As a plumber, your job is to control the excess "flooz" being produced by the Acme Chemicals Co.

Battle Squadron 88

by Tony Preston
Your mission is to eliminate all Barraxian life forms and rescue your comrades.

COLUMNS

New Products 8

The game of Harmony, T.F.M.X. Soundtool, and the Amiga TouchDriver system.

Snapshot 23

by Brad Andrews

A look at Broderbund's Typhoon Thompson, Accolade's Day of the Viper, and more.



**Snapshot-
Virgin Mastertronic
takes on another
Parker Brothers
game, Risk.**

BugBytes 29

by John Steiner
A new VirusX upgrade, some bugs with WordPerfect, and still no answer from Intuitive Technologies.

PD Serendipity 74

by Aimée B. Abren
Highlights include virus checkers, an encoder program, and a 200-page C Manual.

Roomers 76

by The Bandito
The rumors are getting hotter as the days get longer.

The Command Line 78

by Rich Falconburg
Another look at a public domain shell program.

C Notes by the C Group 82

by Stephen Kemp
Linked lists & doubly linked lists in C.



Commodore President Harry Copperman speaks at World of Amiga, New York.

DEPARTMENTS

Editorial 4

Letters 6

Index of Advertisers 80

Public Domain Software 93

COVER PHOTO by
Ernest P. Viveiros, Sr.

Amazing COMPUTING™ / AMIGA

Amazing Computing For The Commodore AMIGA™

ADMINISTRATION

Publisher: Joyce Hicks
Assistant Publisher: Robert J. Hicks
Circulation Manager: Doris Gamble
Asst. Circulation: Traci Desmarais
Corporate Trainer: Virginia Terry Hicks
Traffic Manager: Robert Gamble
International Coordinator: Donna Viveiros
Marketing Manager: Ernest P. Viveiros Sr.
Programming Artist: E. Paul

EDITORIAL

Managing Editor: Don Hicks
Associate Editor: Elizabeth Fedorzyn
Hardware Editor: Ernest P. Viveiros Sr.
Technical Editor: J. Michael Morrison
Video Consultant: Frank Mc Mahon
Copy Editor: Aimée B. Abren
Copy Editor: Derek J. Perry
Copy Editor: Greg Young
Art Director: William Fries
Photographer: Paul Michael
Illustrator: Brian Fox
Research & Editorial Support: Marilyn Gagne
Production Assistant: Melissa-Mae Viveiros

ADVERTISING SALES

Advertising Manager: Donna Marie

1-508-678-4200
1-800-345-3360
FAX 1-508-675-6002

SPECIAL THANKS TO:

Buddy Terrell & Byrd Press
Bob at Riverside Art, Ltd.
Swansea One Hour Photo
Pride Offset, Warwick, RI
Mach 1 Photo

Amazing Computing™ (ISSN 0886-9480) is published monthly by PiM Publications, Inc., Currant Road, P.O. Box 869, Fall River, MA 02722-0869.

Subscriptions in the U.S., 12 issues for \$28.00; in Canada & Mexico surface, \$36.00; foreign surface for \$44.00.

Second-Class Postage paid at Fall River, MA 02722 and additional mailing offices.

POSTMASTER: Send address changes to PiM Publications Inc., P.O. Box 869, Fall River, MA 02722-0869. Printed in the U.S.A. Copyright© May 1990 by PiM Publications, Inc. All rights reserved.

First Class or Air Mail rates available upon request. PiM Publications, Inc. maintains the right to refuse any advertising.

Pim Publications Inc. is not obligated to return unsolicited materials. All requested returns must be received with a Self Addressed Stamped Mailer.

Send article submissions in both manuscript and disk format with your name, address, telephone, and Social Security Number on each to the Associate Editor. Requests for Author's Guides should be directed to the address listed above.

AMIGA™ is a registered trademark of Commodore-Amiga, Inc.

KNOWLEDGE MACHINE

Discover the power of the Amiga with

ReSource™

Intelligent Interactive Disassembler for the Amiga Programmer

- ✓ ReSource will enable you to explore the Amiga. Find out how your favorite program works. Change annoying features. Examine your own compiled code.
- ✓ Load/save *any* file, read disk tracks, or disassemble directly from memory.
- ✓ Automated symbol creation:
JSR -\$1E(A6) becomes JSR _LVOOpen(A6)
MOVE.L #\$3EE,D0 becomes MOVE.L #MODE_NEWFILE,D0
Virtually *all* Amiga symbol bases supported.
- ✓ Now supports user defined symbol bases!
- ✓ Single-key forward and backward referencing makes following subroutines easy!
- ✓ Special support for base-relative addressing.
- ✓ Many files may be successfully reassembled directly from ReSource output. In a trial disassembly, "Preferences" was disassembled, and the resulting source code assembled into a working program, all in under 15 minutes.
- ✓ If you're serious about disassembling code, look no further!

"ReSource is fully-featured and flexible... Everything is fast. The program is astonishing in many ways. The massive size of its internal tables boggles the mind. I admire the remarkable accuracy with which it makes intuitive guesses at the nature of certain bytes."
—Jim Butterfield, Transactor Vol. 2 #5

Now shipping ReSource V4.00 - Order yours now!

VISA, MasterCard, check or money order accepted - no CODs. Not available in retail stores.

Circle 168 on Reader Service card.



The Puzzle Factory, Inc.

P.O. Box 986
Veneta, OR 97487

Orders: (800) 828-9952

Customer Service: (503) 935-3709

\$95



EDITORIAL CONTENT

Power Users, Jets In The Sky

ON A CLEAR AFTERNOON, IF YOU look up at the sky, you can see the vapor trails of a passing jet. You may not see the jet clearly, sometimes you never see the jet at all, but you do see the disturbance it leaves in the upper atmosphere. You know the jet is there by the change it has made.

In much the same way, we sense the growing number of Amiga power users by their level of interest in software and hardware. Power users show their activity by purchasing accelerator cards, hard disks, and advanced versions of the latest software. We know they exist because they create elegant graphics and great programs that appear on public domain bulletin board systems. Yet, very few Amiga users would say they are power users. Why are these users so elusive?

One answer is that often the really busy Amiga user has little time to fill out reader response cards, or write letters to a magazine. These people are busy creating on their Amigas. They pursue a challenge on the Amiga and rarely take the time to let anyone know what they have already accomplished. The other answer is that all of us are power users.

Being A Power User

Each of us exploits the Amiga in one way or another. Each of us pushes the Amiga a little further in a particular field, direction, or use than our fellow Amiga owners. Some of us enjoy word processing on the Amiga and take our favorite word processor to the limit. Some users cannot distinguish CLI from BASIC, but are superb artists and create excellent graphics. Other users program, but do so out of an intellectual curiosity rather than a way to make millions of dollars creating the Great American Arcade Game. Some Amiga owners push themselves and their Amigas in many different directions.

The Amiga is a tool. We chose this tool to achieve our own goals. We purchased our Amiga through a conscious choice. There was no one standing on the sidelines directing us to buy this computer

because it was what everyone else was using. We weighed the factors for and against the Amiga and decided the Amiga would do what we required better than any other computer.

Diversity Is The Issue of This Issue

We termed this issue a "power user" issue because of the nature and diversity of the articles we have included. For the programmer, there are three separate program listings in addition to our regular columns. For the technically inclined, there are two hardware projects. We have included tutorials and reviews on productivity software and graphics. And there is coverage of World Of Amiga, the launch of the Amiga 3000, insight into the still incomplete Workbench 2.0, and more. In fact, we have made this issue as diverse as possible to once again underscore our level of commitment to Amiga power users worldwide. Now, we must request information from you.

Amiga Census '90

The major problem we have as a publication is discovering all the things our readers have done with their Amigas. We are continually searching for the stories and experiences that make computing on the Amiga different from that done on any other machine.

Amazing Computing has always maintained that a computer magazine should be both an information source and a source of inspiration. This inspiration is presented not by saying how great the Amiga is, but by demonstrating the Amiga's greatness. We have never had a better opportunity to do this than now. With the advanced Amiga 3000, the Amiga 2000 with specialized hardware, and the Amiga 500 currently equipped with everything from hard drives to PC-emulation hardware (see the "World Of Amiga" article on page 47 for Pulsar's new announcement), the Amiga needs no frantic flag waving. It needs reporting. This is what AC was created to do and it is what we have always done.

However, we must determine how you, our readers, are using the Amiga. There are two ways in which we can discover how Amiga users utilize the Amiga.

First, you can write us and tell us what you are doing. If you have discovered something wonderful on the Amiga or have used the Amiga in an interesting way, please write us. We spend most of our time searching for the unusual or important ways in which the Amiga is being used today. The best way to develop better users on the Amiga is to let them know how far the Amiga has come, and where it is headed next.

Our second option is to ask you questions. If you subscribe to AC, you may be one of the randomly selected readers who will soon receive a few questions from us on how you use your Amiga. Please answer. Your answers will be combined with those of hundreds of other Amiga users to gain an insight into the Amiga market and its needs.

Although you may be tired of supplying similar information to the Census Bureau, this information is also important. AC needs your responses to consistently provide a publication useful to you. Your answers will help Amiga vendors design and supply the type of Amiga products you need most. And you will show us how you, our Amiga power users, are extending the Amiga's capabilities.

People often ask you how you are or what kind of day you had and never stop to listen to your answers. Now you have an opportunity to speak with someone who is not only interested in your views, but reliant upon them.

Sincerely,

Don Hicks
Managing Editor

Now You
Can

imagine

A Professional 3D Animation Rendering System for the Amiga

For More Information Call ***impulse*** 6 1 2 • 5 6 6 • 0 2 2 1
inc.



Feedback

Dear AC:

In response to Karl D. Belsom's article "Bridging the 3.5" Chasm" in April 1990's issue, there is an easier way of using "AREAD" to transfer files in mass from an Amiga storage device to a Bridgeboard storage device.

First you place all the files you wish to have transferred in one directory. Next you use either PKAZIP or LHARC for the Amiga to compress them into one file. Then use the MS-DOS command "AREAD" with the /b option (to keep the files full binary structure) to copy the file to the MS-DOS storage area of choice. Then use the MS-DOS version of either PKUNZIP or LHARC (depending on which one you used to compress with) to decompress the file into its original files.

PKAZIP or LHARC for the Amiga are compatible with their MS-DOS counterparts LHARC or PKAZIP/PKUNZIP (the latter is two files in MS-DOS). With this process it is simpler to run mass files across the Bridgeboard than it is by scripting. It is also just as easy to transfer files from the Bridgeboard to the Amiga in this manner using "AWRITE". An additional version of the file is that it would be considerably easier to send the information to another computer by modem or networking.

Sincerely,
Brian E. Cucksee
Huntsville, AL

Dear AC:

Your latest issue (April '90) was outstanding! It's not often that I see a hardware project in a magazine outside of *Popular Electronics*, and this issue had TWO, to boot!

Which brings me to the "Biofeedback" project by John Iovine. First, I would like a clarification on the type of Op-amp used in the project, such as "741" or whatever. Also, you might point out to your readers that the US dime that is found in pocket change is **not** made of silver. About the only dimes made of silver these days are found in collections, or high-grade solder. However, they are made of a copper-nickel alloy that might do the job. If silver is needed, one can get silver dimes (well-worn, but just fine for our purposes) for a couple of bucks at one's local coin dealer.

One last thing: for those of you who aren't programmers, here are the answers to Stephen Kemp's questions in April's "C Notes":

1. Yes.
2. African or European?
3. Who's asking?
4. One, but Who's counting.
5. Yes, and the Vulcan Death Grip, too.
6. I told you once.
7. What ever you want it to.
8. Male or Female?
9. No, but I read the books.
10. None. Programmers aren't afraid of the dark.

See you in the funny papers!
Jeff Grimmett
San Diego, CA

John Iovine used a single-ended 5 volt Op-amp (by Images Corp.) for his project. You cannot use an Op-amp that requires a bipole power-supply such as the "741". According to John, any dime or penny can be used, but the dime works best because of its silver content. (Although their silver content is quite small, US dimes do indeed contain silver.) While nickels and quarters do contain some silver, dimes fit in the hand more comfortably. —ED.

Dear AC:

You are truly Amazing! While I generally do not write to magazines, I was going to force myself to just to ask if you wouldn't devote an issue to comparing music sequencing programs. I am in the market to purchase one but couldn't afford to buy all of them.

Well what do I get in the mail but your March issue comparing all the sequencers. Thank you, Thank you, Thank you!

A couple of questions remain. Will there be an upgrade for MusicX? Master Tracks Pro sounds great especially if Passport will eventually port Encore to the Amiga. Is this in the works?

Keep up the good work.
Sandy Thompson
Safety Harbor, FL

Regarding any planned upgrades to MusicX, Microillusions stated, "If you send in the warranty card you will be informed of the upgrade" planned for this summer. We also called PassPort

(continued on page 13)

CanDo!

With CanDo You Can Do Just About Anything!

You can make CanDo your own personal programmer. Just as you no longer have to be a professional artist to create "Art", well now, you don't have to be a programmer to create professional level applications! All you need is CanDo. Because creating a program with CanDo is as easy as drawing a picture with a paint package.



Just Point...Click...Use.

With CanDo, programming is incredibly easy. Programs can be created by simply pointing and clicking. CanDo's graphic interface guides you easily, step-by-step, through the creation of your own applications.



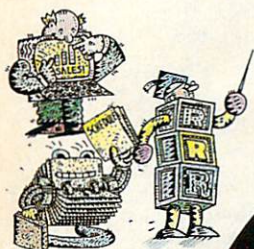
CanDo is a multimedia synchronizer.

CanDo can choreograph digitized sound...music...graphic or video images in real-time, by interaction or time.



CanDo is an animator.

CanDo lets you use any number of Deluxe Paint III brush animations and display them on-screen simultaneously.



CanDo has muscles. And brains, too.

You can use CanDo to create all kinds of applications...from presentations to education to personal productivity, complete with buttons, menus, windows and much, much more.



CanDo has expandable power.

Whether you're a beginner or advanced user, CanDo's powerful scripting language allows you to create the applications you want.



CanDo is compatible.

CanDo utilizes the files generated by a host of popular software packages. It works with all Amiga resolutions, digitized sounds...and even supports ARexx.

With tools like these, you really can do just about anything. The only limitation is your imagination.



Now You Can Win a FREE Trip To the Amiga '90 show in Cologne, West Germany. See Your Participating Dealer For Details.

CanDo is a product of Inovatronics, Inc.
8499 Greenville Avenue Suite 209B Dallas, Texas 75231 214-340-4991
Dealers and Distributors should contact: Southern Technologies, 800-647-7741
DELUXE PAINT III is a trademark of Electronics Arts, Inc.; ARexx is copyrighted by William S. Hawes.

Circle 110 on Reader Service card.

NEW PRODUCTS & other neat stuff

SOUND ADVICE

Now being distributed by Imtronics is the *T.F.M.X. Soundtool*, a new music editor from Demonware of West Germany. Consisting basically of two parts—the editor (to input the music data), and the music routine (to play back the music data)—the program is designed to provide users with the ability to prepare complex musical pieces and sound effects and to integrate them into their own programs.

Among its many features, the *T.F.M.X. Soundtool* program provides an intelligent note and control panel, as well as special effects such as vibrato, portamento, and echo.

The Imtronics package includes 2 samples disk in addition to the program disk. The *T.F.M.X. Soundtool* runs on the A500, 1000, and 2000, with a minimum 512K memory required. *List Price: \$79.95. Imtronics, 12301 South West 132 Court, Miami, FL 33186, (305) 255-9302. Inquiry #261.*



The T.F.M.X. Soundtool package.

SAFETY IN NUMBERS

HelpKey Development, a new entry in the Amiga market, recently released *The Buddy System For DeluxePaint III*, a two-disk set containing over 90 lessons that take the user through the many features of Electronic Arts' *DeluxePaint III*.

By taking advantage of the Amiga's multitasking capabilities, *The Buddy System* can be run at the same time as *DPaint III*, allowing the user to switch back and forth between the tutorial interface and *DPaint III*. To find information, simply select the appropriate icon or menu item. Each selection reveals a complete lesson that can include text information, a descriptive picture, and/or a guided demonstration. *The Buddy System* also takes advantage of the Amiga's built-in speech capabilities, with spoken narration accompanying each lesson.

All areas of the *DPaint III* program are covered, from the basic functions of the drawing tools to the advanced techniques of animation. Demonstrations of many practical techniques, such as use of textures, are also included.

The Buddy System For DeluxePaint III runs on the A500, 1000, or 2000, with 1 megabyte minimum memory and KickStart v1.3 required. *List Price: \$49.95. HelpKey Development, 6671 West Indianatown Road, Suite 56360, Jupiter, FL 33458, (407) 694-1756. Inquiry #262.*



PRESENTING

WORLD OF AMIGA IN CHICAGO

★ *Starring* ★

THE AMAZING AMIGA

★ *Featuring* ★

**Amiga Hardware • Amiga Software
Amiga Accessories • Seminars • Bargains**

**Rosemont O'Hare Expo Center,
Rosemont, Illinois**

October 5-7, 1990

Friday, Saturday & Sunday 10am-5pm

Pre-registration:

\$8 per day or \$20 for 3 days

Deadline for pre-registration Sept. 18

Registration at show:

\$10 for 1 day

\$25 for 3 days

Registration includes exhibits and seminars.



**WORLD OF
AMIGA**
IN CHICAGO

Produced by
World of Commodore
Shows, The Hunter
Group

For more information call (416) 595-5906 or fax (416) 595-5093.

SAVE WITH PRE-REGISTRATION

Please register me for the 1990 World of Amiga in Chicago at the Special Pre-registration Rate.

NAME _____

COMPANY (if applicable) _____

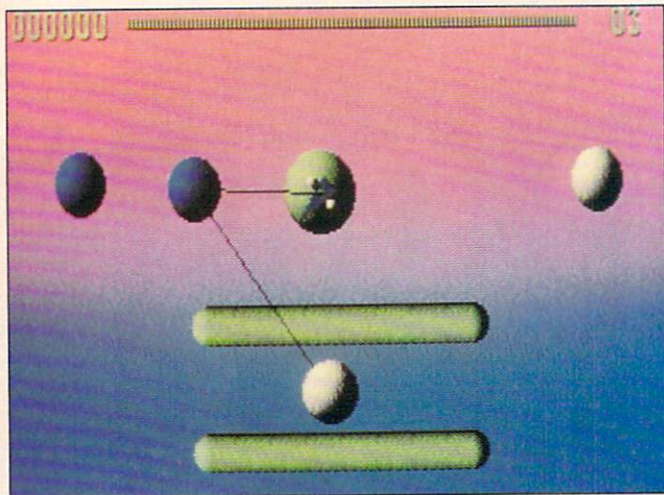
ADDRESS _____

CITY _____ STATE _____ ZIP _____

☐ One-day registration (\$8) ☐ Three-day registration Oct.5-7 (\$20)

Make Check or Money Order payable to
The Hunter Group, 3380 Sheridan Drive, Suite 120, Amherst, NY 14226

Circle 111 on Reader Service card.



Harmonizing with Accolade

INSTANT KARMA

Earth Day, dolphin-free tuna, and now this—a game the object of which is not to annihilate objects, but to keep them from destruction. Such is the premise of **Accolade's The Game of Harmony**, a game where players combine strategy and concentration in an effort to maneuver like-colored orbs together before they explode with tension.

The game contains fifty progressively challenging levels randomly arranged with orbs which glide around the screen while you attempt to gain points, and eventually

achieve ultimate harmony, by gently pushing pairs towards each other with a pointer, or "the seeker". Add to this non-violent image a New Age musical score determined by the random touch and actions of objects on a given level. If you listen very closely, you can hear Rambo dying a slow death.

The Game of Harmony will run on the A500, 1000, or 2000, with a minimum 512K memory required. **List Price: \$44.95. Accolade, 550 South Winchester Boulevard, Suite 200, San Jose, CA 95128, (408) 985-1700. Inquiry #263.**



West Hollywood, CA's "Trendy Melrose Ave." is indicated on the touch screen driven videodisc application at the LA International Airport. The system is run on an Amiga that has a MicroTouch screen installed on it.

GOLDEN BEAR ABROAD

Accolade has released the third entry in their series of golf simulations created in association with Jack Nicklaus Productions. **Jack Nicklaus Presents the International Course Disk**, designed exclusively for use with *Jack Nicklaus' Greatest 18 Holes of Major Championship Golf*, takes the computer golfer abroad to tee off on precise simulations of three championship courses designed by Jack Nicklaus including the Australian Golf Club in Kensington, Australia, the Saint Creek Golf Club in Nagoya, Japan, and the St. Mellion Golf & Country Club in Cornwall, England.

The International Course Disk will run on the A500, 1000, or 2000, with a minimum 512K memory required. *Jack Nicklaus' Greatest 18 Holes* is also required. **List Price: \$21.95. Accolade, 550 South Winchester Boulevard, Suite 200, San Jose, CA 95128, (408) 985-1700. Inquiry #264.**

REACH OUT AND TOUCH AN AMIGA

MicroTouch Systems, Inc. has become the first touch screen manufacturer to announce a fully integrated touch screen system for the Amiga. The **Amiga TouchDriver** system consists of the MicroTouch Screen (the company's high resolution, analog capacitive touch screen), driver software and optional monitor. The driver enables all mouse-driven software to work with touch input.

The *Amiga TouchDriver* is the first to provide a two-button mouse emulation touch screen for the Amiga. With the *Amiga TouchDriver* users can, for the first time, use a touch screen for any Amiga application. Users can select from menus, create drawings, and manipulate

SOMETIMES YOU GOTTA FOLLOW THE RULES

For C programmers out there looking to push **Blue Ribbon Bakery's Bars&Pipes** to the outer limits of music composition, **Rules for Tools** may prove key. This comprehensive guide to the ins and outs of Bars&Pipes is designed to help users create their own Bars&Pipes tools from scratch. *Rules for Tools*, which is part of the *Bars&Pipes Add-on Series*, includes over 100 pages of documentation along with numerous source code examples.

To order *Rules for Tools*, send a check or money order for \$49.00 (\$44.00 plus \$5.00 shipping and handling) to **Blue Ribbon Bakery, Inc., 1248 Clairmont Road, Suite 3d, Decatur, GA 30030. Inquiry #265.**

windows simply by touching the screen—without reaching, orienting, or clicking a mouse. Fully compatible with the Amiga mouse, the driver also offers users an expanded Preferences menu for greater button and cursor control definition.

The all-glass MicroTouch Screen has a high resolution of 1,024 x 1,024 touch points, allowing the user to easily point to any size icon or menu item and to locate the cursor down to the pixel level.

The *Amiga TouchDriver* site license lists for \$395, and the complete MicroTouch Screen with controller sells for as little as \$350.00 in Original Equipment Manufacturers (OEM) quantities. **MicroTouch Systems, Inc., 55 Jonspin Road, Wilmington, MA 01887, (508) 694-9900. Inquiry #266.**

OVER

100,000 Satisfied Customers

NOW EVEN LOWER PRICES!!!

MEMORY

M501 A500	79.95
Starboard 1MB	289.95
Starboard 2MB	399.95
8-UP with 2MB	289.95
8-UP with 4MB	449.95
8-UP with 6MB	599.95
8-UP with 8MB	749.95



System Packages

We customize AMIGA 2000 and AMIGA 3000 desktop video systems to meet your individual needs. Call and talk to one of our system specialists to get the best price for your requirements.



GREAT VALLEY PRODUCTS INC.

28 MHz 68030	33 MHz 68030
28 MHz 68882	33 MHz 68882
with 4 MB of 32 bit memory	with 4 MB of 32 bit memory
1949.95	2549.95
with 40 MB hard drive	with 40 MB hard drive
2399.95	2899.95
with 80 MB hard drive	with 80 MB hard drive
2699.95	3199.95

HARD DRIVES

GVP Hardcard /0	159.95
GVP A2000-2/0	179.95
GVP A2000-8/0	199.95
HardFrame	219.95
Hardcard 40MB	499.95
Hardcard 80MB	699.95
Hardcard 105MB	799.95
Quantum Pro 40	379.95
Quantum Pro 80	649.95
Quantum Pro 105	699.95
Supra 105/1000	1099.95
Supra 105/500	999.95
Supra 40/1000	683.95
Supra 40/500	599.95
Supra 80/1000	949.95
Supra 80/500	875.95
Supra Wordsync	169.95

VIDEO

Flicker Fixer	469.95
Framebuffer with 1MB	749.95
Magni 4004S	
with remote	1629.95
MiniGen	189.95
Panasonic 1410	209.95
Panasonic 1500	299.95
Panasonic Vari-Lens	39.95
Polaroid Freezeframe	1599.95
ProGen	349.95
Sharp JX-100	749.95
SuperGen	669.95
SuperGen 2000	1699.95

SOFTWARE

Deluxe Paint III	89.95
Digi Paint	64.95
Digi View 4.0	129.95
Digi Works 3D	79.95
Diskmanager MAC	79.95
Homebuilder's CAD	119.95
Lattice ++	249.95
Pagestream 2.0	179.95
Prowrite 3.0	99.95
Saxon Publisher	249.95
Soundscape Pro MIDI	109.95
Turbo Silver	109.95

CALL FOR UNLISTED TITLES

FDATA-10
99.95

FDATA-20
239.95

HP PAINTJET
1099.95

Government and School Purchase Orders Accepted.



InterComputing, Inc.

2100 N Hwy 360, Suite 2101, Dallas, TX 75050-1015

Circle 116 on Reader Service card.

1-800-622-9177

Customer Service: 214-988-3500

InterComputing Deutschland Inc.

Schönbecker Str. 55-57
5600 Wuppertal-2

Telefon: 0202/89155
Telefon: 0202/89304

InterComputing France

34, Avenue des Champs Elysees
75008 Paris

Phone: (1) 42821603
FAX: (1) 42806649

As always we have the most 'customer friendly' terms: SIH \$4.95 in cont. USA; \$30.00 min. order; MASTERCARD & VISA with NO credit card fee; in Texas add 7% Sales Tax. \$12.00 shipping to APO/FPO addresses. RMA# required on all returns. All prices subject to change.



AMIGA • AMIGA CHIPS, PARTS & UPGRADES • AMIGA			
•Fatter (Super) AGNUS 8372 -\$99.50 with simple 10 minute step by step instructions, FREE Chip/Extractor valued at \$15.95 at no charge.			
• A501-512 RAM	\$89.50	• 256 x 4/100	\$12.50
• 1.3 Kickstart ROM	\$27.95	• 1 MEG x 1/100	\$10.40
• 8520	\$17.95	• 8362/64	\$49.95
• A-500 H/D Power Supply	\$73.50	• A-2000 Power Supply	\$149.00
• Amiga Diagnostician #7	\$11.95		
• AMIGA 1000 REJUVENATOR PACKAGE •			
New Product - The Amiga 1000 Expansion Board is now available with the following features: Utilizes the Fatter Agnus Chip, 1.3/1.4 Kickstart ROM and New Denise • One Meg of Chip RAM • Clock-Battery Backup • Simple Solderless Installation • 100% Compatibility with all Products/Software • Various Packages Available • \$479.00 complete.			
		The Grapevine Group, Inc. 35 Charlotte Drive Wesley Hills, NY 10977 1-800-292-7445	
Prices subject to change		• (914) 354-4448 FAX (914) 354-6696 Send for complete catalog	

Circle 147 on Reader Service card.

TRAVELIN' MAN

New from **Interplay Productions** comes **Future Wars: Adventures in Time**, a graphic time travel adventure game in which the main character has been hurled back into the Middle Ages. After claiming his luggage, he discovers an intergalactic plot the horror of which threatens to transcend centuries. Your future (not to mention mankind's) depends on your ability to crisscross through time, solve an assortment of puzzles, and battle the various mutant monsters and alien storm troopers that one tends to befall during extended periods of time travel.

Future Wars: Adventures In Time is the first release in the new "Cinematic" system of graphic adventures, whereby the interface utilizes a series of pop-up text and command windows, thus eliminating the need for a parser and making the game ultra user-friendly.

Future Wars: Adventures In Time runs on the the A500, 1000, 2000, and 2500, with a minimum 512K memory and Kickstart v1.2-1.3 required. **List Price: \$49.95.** *Interplay Productions, 1575 Corporate Drive, Costa Mesa, CA 92626, (714) 545-9001, Inquiry #267.*

PAWNING OFF

Eagle Tree Software has brought Chinese Chess to the Amiga. **Chinese Chess, The Science Of War** features the ancient cousin of Western chess in multiple 2D and 3D views, complete with sinuous river that divides the game board. The game includes several levels of difficulty, clocks, on-line descriptions of rules, pieces, notation and strategy, the ability to print, save or load games, as well as the ability to create custom positions. Pieces can be controlled with either the mouse or the keyboard. *Chinese Chess* will run on the A500, 1000, or 2000, with a minimum 512K memory required. **List Price: \$32.95.** *Eagle Tree Software, P.O. Box 164, Hopewell, VA 23860, (804) 452-0623, Inquiry #268.*

TOSS THE TELEPROMPTER

Would-be talking heads rejoice! Now available from Australia-based **DigiSoft** is **AutoPrompt**, a scrolling script prompter and text editor program designed for television script prompting and information displays.

The DigiSoft program is comprised of two interleaved components, Edit mode and Prompt mode, with the ability to swap between the two modes at will provided. Using the Edit mode, text can be typed, imported, edited, saved to disk, loaded, and re-edited as required. Scripts from other word processors or text editors can also be imported. In the Prompt mode, the text is displayed. The Prompt mode features user-selectable fonts, and either the cursor keys, mouse, or joystick can be used to control the speed and direction of scrolling.

AutoPrompt supports PAL and NTSC in both interlaced and non-interlaced modes, and high or low resolution screens. The program also features a full intuition interface complete with menus and keyboard shortcuts.

AutoPrompt runs on the A500, 1000, 2000, or 2500, with a minimum 512K memory and Kickstart v1.2 or later required.

List Price: \$295.00. *DigiSoft, 12 Dinmore Street, Moorooka, Brisbane 4105, Queensland, Australia, 61-7-277-3255, Inquiry #269.*

•AC•

Barney Bear Goes To School
 List Price: \$34.95
Free Spirit Software
 P.O. Box 128
 58 Noble Street
 Kutztown, PA 19530
 (215) 683-5609
 Inquiry #270

688 Attack Sub
 List Price: \$54.95
Electronic Arts
 1820 Gateway Drive
 San Mateo, CA 94404
 (800) 245-4525
 Inquiry #271

The Legend of William Tell
 List Price: \$39.95
Electronic Zoo
 3431-A Benson Avenue
 Baltimore, MD 21227
 (301) 646-5031
 Inquiry #272

Other Products Received

SPOC DIGEST

Magazine-type demo disk

Useful programs, fun and games from around the world plus interesting articles, news and ideas for your AMIGA. *Contains much more than magazines costing \$15.00!* Also, info on our **SPOC DISK**, along with free programs from this disk. Just send **\$5.00** to help cover the cost of this ad to:

**SPOC
BOX 299
KIOWA, OK 74553**
Circle 112 on Reader Service card.

(continued from page 6)

Design, Inc. in regards to your inquiry about Encore. East Coast Regional Sales manager Joel Heppting said Passport is now developing Encore for the IBM and Atari ST. The next development will probably be for the Amiga.—ED.

Dear AC:

In the April, 1990 issue of *Amazing Computing*, a review by Cletus Baker of the telecommunications program Baud Bandit contains the following quote:

"To date, only Online! (and its brethren Platinum and The Works versions), and the shareware JRComm feature ZModem."

The quote is not simply inaccurate but blatantly false, and just a little more research on the part of your reviewer would have easily discovered the full facts. The telecommunications program A-Talk III, produced by our company, Felsina Software, published by OXXI, Inc., has included the ZModem protocol since its initial release in September of 1988. The mistake is even more surprising considering that A-Talk III has received numerous awards including Innovations '89 at Summer CES in Chicago and Expert Choice Award '89 from both AmigaWorld and Amiga Plus magazines. Interestingly enough, while *Amazing Computing* reviewed A-Talk Plus back in 1988, no review of A-Talk III has ever appeared since, while you seem to be able to review every possible update of Online! I'm sure my publisher will be able to furnish you with a complimentary copy of A-Talk III, so

you can check your facts firsthand. Thank you very much for your time.

Sincerely,
Marco Papa
Felsina Software

Dear AC:

Imagine our surprise to read in Cletus Baker's review in your April 1990 issue that only Online and Baud Bandit commercial terminal programs and the shareware JRComm have the ability to do file transfers with the ZModem protocol. This will come as a shock to thousands of A-Talk III users who have long been transferring files with ZModem, ZModem Resume and ZModem Remote, and doing ZModem batch transfers of binary files with a simple, menu-supported command.

I assume that in Baker's search for the terminal program, by examining mostly freely-distributable software and "the more expensive commercial versions", he simply overlooked A-Talk III. In fact, every feature he praises in Baud Bandit is matched or surpassed by A-Talk III.

For users who have been misled by this inaccuracy in Baker's article, I suggest they look instead for a full-featured telecomm program which offers them not only ZModem, but a complete range of standard, XPR and shared-mode protocols, more than one or two terminal emulations, support for all the common modem types (including the new super-high-speed modems), remote hosting abilities, multi-serial port compatibility, and a complete library of ready-made scripts, including one for running a BBS. When they find all that in one package for less than \$100.00, their search for the perfect terminal program will be over.

Sincerely,
Patricia Cummings
Director of Technical Support
OXXI, Inc.

—We contacted Cletus Baker and here is his response:

"As Ms. Cummings suggests, I simply overlooked A-Talk III in my search for 'The Perfect Terminal Program'. My experience encompassed the most-often-discussed programs among my friends and associates, and in our conversations A-Talk III never came up. I must apologize to Ms. Cummings and Mr. Papa for my lack of awareness."—Cletus Baker

Dear AC:

I am having fun with Robert D'Asto's Ham Bones, the Ham_Toy program on page 73 of the April 1990 issue of *Amazing Computing*. As he states, there is no provision for saving a picture that you have drawn. I am using GRABBIT, Discovery Software, [to] print or save any screen, from any program anytime. I can save a screen drawn with Ham_Toy and save it with GRABBIT, then I load the screen into DIGI PAINT, New Tek Incorporated. Once loaded into DIGI PAINT, I can manipulate the picture whichever way pleases me.

I dislike using a program where one spends a lot of time drawing a picture and then cannot keep it or work on it at another time. [My] suggestions above make the Ham_Toy program more worthwhile.

Sincerely,
Herbert H. Starkey
Salem, OR

—We have tried to contact Discovery Software since late last year, but the phone has been disconnected. Since we are unsure if GRABBIT is still available (or if Discovery Software itself is viable) we decided not to suggest using the program to save Ham_Toy pictures.—ED.

Dear AC:

In your April issue [...] there is a Bio-Feedback/Lie Detector Device article which says that there is an included program that I cannot locate. I would appreciate a reply in this matter. Thank you.

Sincerely yours,
Michael E. Davis
St. Paul, MN

—The Listing was inadvertently omitted in April. We did publish it in the May issue of AC, on page 91.—ED.

All letters are subject to editing. Questions or comments should be sent to:

Amazing Computing
P.O. Box 869
Fall River, MA 02722-0869
Attn: Feedback

Readers whose letters are published will receive five public domain disks FREE.

BY JOHN STEINER

PageStream 1.8

PUTTING SOFT-LOGIK'S DESKTOP PUBLISHING PACKAGE TO WORK

NEARLY ALL OF THE FEATURES THAT SHOULD BE IN A TRULY PROFESSIONAL desktop publishing package are in Soft-Logik's PageStream 1.8. If you are like me and do a lot of desktop publishing, you will no doubt appreciate its many advanced features.

The best way to test software is to actually put it to work. With that in mind, several different kinds of documents were created similar to those I normally work with, to see how quickly and intuitively PageStream operates. The major features discussed herein are those I had an opportunity to work with using these test documents. Tests were done on an Amiga 2000 with 3 MB of RAM and a hard disk and 2090A controller card installed.

To begin, the feature that permits adjustment of paragraph indents and "outdents" (i.e., hanging indents) is intelligently designed, and very easy to use. A graphic display shows exactly where the new setting is before you have to commit to it. After all, nobody likes to try a setting and click OK, then realize it is not right when the page is redrawn. The PageStream requester lets you see that it's right before you close the requester.

Full adjustment of character tracking, kerning and line leading are provided for. Text height and width is measured in points, and can be set anywhere from 1/50 of a point to 1310 points! I had some problems when working with text over 750 points tall, but how often does a desktop publisher work with text so large that each letter cannot fit on an 8.5-inch by 11-inch page? The claim that you can work with text as large as 1310 points may be accurate, but you must have large enough paper. You can even set the point size different for horizontal and vertical pixels. I printed a sign with characters 200 points wide and 450 points tall.

Paragraph tagging is implemented in a very workable fashion. Tags can be applied to paragraphs, making the layout of large documents fast and easy. There does appear to be one bug related to tagging of text objects. During one test, I had a problem with disappearing text. Changing

the tag seemed impossible. Other boxes I chose did not exhibit the problem. I also found that once text is tagged, you apparently cannot change text attributes without either changing or removing the tag, which is as it should be (although there wasn't any warning that this was happening). Some confusion ensued when I tried to change a block of previously-tagged text without changing the tag itself, and the text wouldn't change. Initially I thought I had encountered some kind of bug, and it would have been nice had a warning popped up to remind me that the text I was trying to modify was tagged.

Manual and batch hyphenation is allowed, and a spelling checker is built-in. Hyphenation can be applied to or removed from text selectively, but you may need to spend a few extra moments with the manual to master the selection buttons for this feature. The spell checking is nice, but I would rather see the program code used to do something else. Spell checking can be more effectively done within the word processor, as the text is being created. Editorial changes made within PageStream would then not be subject to spell checking by the computer—but we all know that editors are perfect spellers anyway!

Any desktop publishing software worthy of merit has to be able to wrap text around graphics. PageStream does, with four modes of text runaround. You can

wrap text around the left, the right, or both sides of structured graphics. Text cannot wrap around irregular-shaped objects in bitmap graphic images, but you can outline the bitmapped image with a structured graphic line that has no color. Then, simply set the text runaround mode to wrap around the invisible structured graphic element.

Programs that are the most powerful also have some capability to automate repetitive tasks. PageStream is equipped with a powerful macro capability that allows you to do complex operations with a single macro entry. Once the macro is designed, you can call it up at any time to perform its complex function. I did not have the opportunity to test this feature except to try a couple of the macros that were demonstrated in the tutorials.

The stereotypical image of the professional desktop publisher includes that of an expensive laser printer connected directly to the computer. The reality is that many desktop publishers look with envy upon laser printer owners, and must print their own documents on a dot matrix printer. PageStream can print on dot matrix printers as well as Postscript laser printers (Postscript is the standard language of high-quality printing devices for desktop publishers). PageStream documents can print at 2500 dots-per-inch on a Postscript-based typesetting machine for a magazine-quality print output. The dot matrix output in earlier versions of the program was not very smooth, but that has been corrected. Even nine-pin output is acceptable for club newsletters and for proofing of documents destined to be typeset or laser printed. You can even dot matrix print in landscape mode (wide rather than tall page orientation, i.e., a page printed sideways) on a dot matrix printer. Once you have proofread a document, you can generate a Postscript file on disk which can be taken to a service bureau for final printout on a laser printer or typesetting machine.

Ed. Note: At World of Amiga in New York, Soft-Logik Publishing Corp. announced PageStream 2.0. For details on this future release, please see our show report on page 47.

PageStream also permits the composition of pages that are actually larger than your printer can handle in a single pass through a feature called "tiling". This option automatically divides each single oversized page into several overlapping areas which are printed onto smaller sheets, cut apart, and pasted together to form one large master page. The person who does your print job can use the large master when putting together final camera-ready copy. The tiling option works for pages that are as large as 18 by 18 inches.

Professional desktop publishing programs should probably be able to handle pages up to full newspaper size; however, and this is not possible with PageStream 1.8. There also doesn't appear to be any way to change the size of a page once it is created. Here, just create a page of the correct size, and move information over from the former page.

Though there are several filters for importing different bitmap graphic images, Amiga owners will be most interested in importing IFF ILBM graphics. IFF bitmap graphics of any resolution (including HAM images) import easily into PageStream, either as complete images or as picture windows which can then be cropped. Depressing the shift key while drawing the box that holds such an image maintains the correct height/width ratio for the picture.

PageStream also does color separations, both mechanical and four color. I printed some four-color separations and they looked fine out of the laser printer, but I did not have the opportunity to view them as a color printed page. PageStream also prints negative and mirror images if the person at your print shop requests them. I did not test these features.

PageStream excels in its layout and page view capabilities. Each page is displayed in a fully-sizeable window, the view can be changed from 200% of actual size to full page, and the variable zoom feature lets you look closely at any part of a page. The Show Full Width feature that displays the entire width of a page is useful. A set of left and right Master Pages allows you to put together a template of items such as headers, footers and page numbers, as well as graphic elements, that will automatically appear on every page. Page numbers can be placed on the master pages, and they will appear on every page in either Arabic or Roman numerals. Page numbers can be set to start from any number desired.

Columns for text can appear in any order on any page. Columns containing text can be linked so the text will automatically flow from column to column, even across page boundaries. A background

grid and page guides can be used to help place critically-positioned columns and graphics. Pages are easily inserted, deleted, and moved.

The structured graphic capabilities of the program can be used to create special effects and limited structured drawings. Lots of fill patterns are available, and you can even create your own pattern if you want. Up to 256 colors can be selected for creating graphics in a given document. I had a little trouble selecting graphic elements that I had purposely sized to be too large to fit on a page. If you accidentally specify a size that is too large, be sure to change it back to fit on the page before you try to select something else. It may take a while to select the object again.

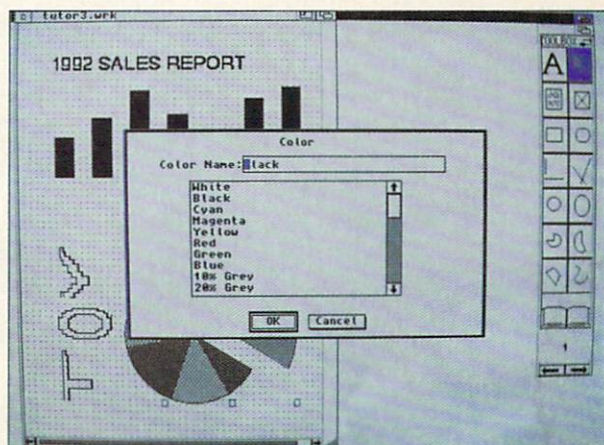
The ability to rotate text and graphics is beautifully implemented. The requester that allows rotation of any object (or group of objects) demonstrates with a rotating rectangle how the final position selected will make the object appear. This feature makes for easy placement of text at any angle.

There is a bug in the rotate function, however. If a text object is rotated, then changed in size, the object is redrawn as a mirror image. Recall the rotate requester, and you find that the rotate coordinates have changed. Simply change them back, and the text returns to where it was originally. To avoid this, be sure to size the text or graphic object before you rotate it.

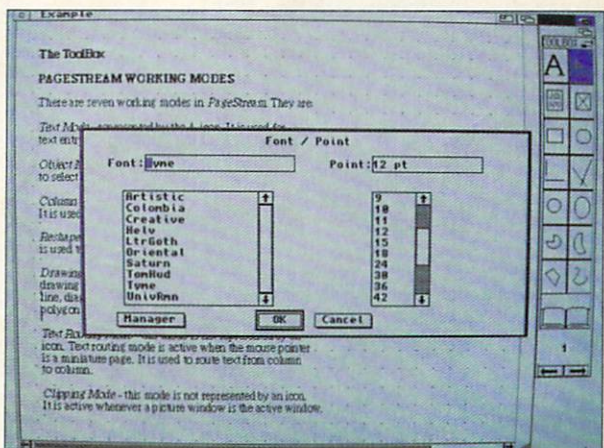
The program is not without problems, but they do seem to be relatively minor in nature, and often related to the philosophical implementation of a desktop publishing interface, rather than problems caused by bugs. This is as it should be. Problems with software implementation should not mean features that don't work, but rather, features that are either missing or that could be implemented in a more intuitive manner. PageStream certainly has no shortage of features, and when you measure the quantity and quality of them against the program's relatively low cost, the program is really a bargain. With this in mind,

let's now look at a few of the shortcomings I discovered.

My biggest complaint is that, for the most part, the fonts you get with PageStream 1.8 are ugly—there is simply no nicer word for them. With the exception of Tyme and Helv (versions of the venerable Times Roman and Helvetica), the fonts generally look amateurish and really give a document a look that screams "This page has been desktop published!" A powerful desktop publishing program should naturally create documents that do not call attention to themselves. Since standard Amiga fonts cannot be used in PageStream documents, and most Amiga fonts aren't meant for publishing applications anyway, a source of good-looking fonts is necessary. Fortunately, Soft-Logik provides many optional font sets (the quality of which I was not able to verify). The font sets are available to drive either Postscript or dot-matrix printers. Third-party compa-



Color selection (above) and font selection (below) in Soft-Logik's PageStream 1.8.



1. Dance on a volcano.
2. Live amongst elves.
3. Read every issue of
Amazing Computing
ever published.

We've just made your life's goals one easier.

No doubt you have over the years compiled a list of goals you hope to achieve. And certainly *Amazing Computing* holds a prominent position on that list. That's why we've made it that much easier for you to acquire the complete *Amazing* library at terrific savings. For a limited time, you may purchase volume sets of AC at
a savings of over 50%!

AC Volume 1 is now available for just

\$19.95*!

(A regular \$45.00 value, this first year of AC includes 9 info-packed issues.)

AC Volumes 2, 3, & 4 are now yours for just

\$29.95* each!

(Volumes 2, 3, & 4 include 12 issues each and regularly sell for \$60.00 per volume set.)

PLUS! We're now offering subscribers freely redistributable disks** at distribution prices. Now's the time to complete your Fred Fish, Amicus, or AC disk collection. Pricing for subscribers is as follows:

1 to 9 disks: \$6.00 each
10 to 49 disks: \$5.00 each
50 to 99 disks: \$4.00 each
100 disks or more: \$3.00 each

(Disks are \$7.00 each for non-subscribers.)

To order volume sets, freely redistributable disks, as well as single issues, use your Visa or MasterCard and

call 1-800-345-3360.

Or just fill out the order form insert in this issue.

Amazing Computing and freely redistributable software—at savings
beyond your wildest dreams.

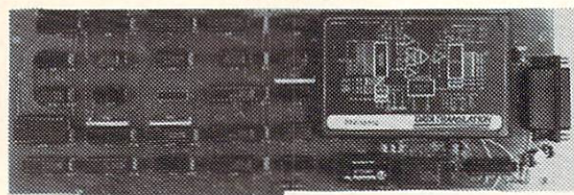
*Postage & handling for each volume is \$4.00 in the US, \$7.50 for surface in Canada and Mexico, and \$10.00 for all other foreign surface.

**AC warrants all disks for 90 days. No additional charge for postage and handling on disk orders. AC issues Mr. Fred Fish a royalty on all disk sales to encourage the leading Amiga program anthologist to continue his outstanding work.

Let ACDA Open Your Real World Window !

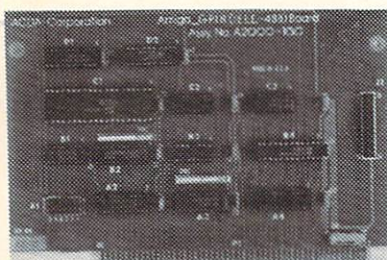
Scientific and Engineering Products for Your Amigas!

PROTO-40K



Proto-40K is the first and only fully featured data-acquisition and process-control expansion card for the Amiga 2000. The Proto-40K features a 16 channel 12-bit multiplexed analog-to-digital converter, two 8-bit digital-to-analog converters, a 3-channel programmable timebase, 16 digital inputs and 16 digital outputs. Proto-40K also features a highly stable instrumentation amplifier with programmable gain, multiple triggering sources, and on-board digital waveform generation. Data acquisition and process control projects are a snap to develop with the Proto-40K Data Acquisition System (DAS) software and 'C' source code. Sample application programs and source are included for each of the Proto-40K functions. Now sold in various custom component configurations. Buy only the functions you need. Call for new lower pricing.

Amiga GPIB



Amiga_GPIB is a General Purpose Interface Bus card for the Amiga 2000. This half-length expansion card performs all the Talker, Listener, and Controller functions of the GPIB (IEEE-488) protocol. One Amiga can control up to 14 GPIB devices. Includes Command Function Library (ACDA GPIB CFL), test application program and 'C' source code driver. \$495.00



DigiScope

DigiScope is a digital storage oscilloscope emulator that works with ACDA's Proto-5K, Proto-40K or other parallel-port digitizers. DigiScope has 16 independent waveform buffers, a digital signal processing (DSP) package, a Fast Fourier Transform (FFT) package and a filtering package. DigiScope has extensive waveform scrolling functions that work in a resizable scope window in high or low screen resolution. DigiScope offers a complete set of archival functions and the standard complement of signal statistics. DigiScope also features an extensive digital waveform generator package. \$139.95

Shinko & Mitsubishi Preferences 1.3 Printer Drivers

We offer a complete line of thermal color printer drivers for the Mitsubishi and Shinko A&B size color printers. They are 100% Amiga Preferences 1.3 drivers. \$133.00

AmigaView 2.0

AmigaView is an object-oriented, C language, Intuition front-end interface library that provides over 100 easy-to-use routines and macros. Our package features WINDOWS, SCREENS, MENUS, REQUESTERS, GADGETS OF ALL TYPES (including automatic mutual exclusion), BITMAPS, ALL IMAGERY, IFF, TEXT, and much more. This standardized and consistent Intuition/Graphics interface greatly reduces programming time and code space for professional applications development. AmigaView works with both MANX and LATTICE. See AmigaWorld (Sept./Oct. 1987, p.28) for review. \$79.95

Amiga FFT C Package

The Amiga FFT C Package Provides all the source you need to perform detailed frequency analysis utilizing a complete set of Fast Fourier Transform (FFT) routines. The package includes C source for derivation of the Power-Spectrum, Phase-Amplitude Spectrum, Inverse FFT, several window functions and user interface functions. \$152.00

ACDA Corporation
220 Belle Meade Avenue
Setauket, NY 11733
(516) 689-7722

Circle 104 on Reader Service card.

nies are also producing font sets for PageStream. There are several public domain PageStream-compatible fonts available on information services such as People Link and Genie. If you have access to a Postscript printer, I recommend Font Disk A from Soft-Logik, as it contains screen versions and font metrics for the family of fonts built into the majority of Postscript printers being sold today. These fonts are attractive and contain enough variety to give your documents a professional look. If you don't have access to a Postscript printer, and you want to duplicate the fonts that come with font disk A, you will find that (unfortunately) they have been spread across several disks. PageStream font disks sold directly from Soft-Logik all sell for a suggested retail price of \$39.95.

I also had some installation problems when I went to put the program on my hard drive. The instructions were clear and easy to understand and I followed them exactly, yet when I was finished, the program did not work properly. It took much effort to get the program to recognize where the fonts and drivers were when the program started up. Each time I ran the program, I had to manually tell the software where to find all the default PageStream system drawers, and it could never find the import filters or dictionaries because the Global

Set/Save Paths option would not work properly. I had put all the PageStream related files and drawers in an FH1:PageStream directory I had created on my hard disk. I found that in order to make PageStream work properly on my system, I had to use the assign command to assign PAGESTREAMFONTS: and PAGESTREAMDRIVERS: to FH1:PageStream before executing the program. Once that was done, the program started and worked as it was supposed to. This problem also occurred when I tried to install the software on a stock 1 MB 2000HD. I visited with both a PageStream beta tester and Soft-Logik technical support about this problem, thinking they had inadvertently left the information about making the assignments out of their manual. Both people told me that no assigns are required, and neither could explain why I was having the problem described here. I even went through the Global Set/Save Paths requester with the person from Tech Support to make sure I had set my paths correctly. He simply didn't have an answer, other than to say that since it works with the assigns they should be left in, even though they are not supposed to be needed.

The program does not use the busy pointer enough to suit me. Normally, when PageStream is busy, it replaces the usual

arrow pointer with an hourglass shape to indicate the program is busy. The problem is that there are times when the program is busy, but the arrow pointer is still active. This usually happens shortly after selecting a menu item, and gives the false indication that either the menu selection didn't work, or the program has locked up.

There is also a problem with the HP LaserJet driver that is included with PageStream. It will not drive my QMS PS 810 Plus printer properly when the printer is in LaserJet emulation mode. The QMS printer is a Postscript printer, and PageStream works just fine with it in the Postscript mode. When I use the LaserJet driver and put the printer into LaserJet mode, I get about 1/16th of a good page, then it ejects and several pages of garbage text follow. There is possibly an incompatibility with the HP emulation in the printer; however, we have used that emulation mode on a regular basis while driving IBM software, and have never before found a program that didn't drive the printer properly. The Amiga Preferences LaserJet driver works fine with the printer also. PageStream does drive other HP LaserJet-compatible printers properly, and I know of several PageStream users who are printing on their LaserJet without problems.

(continued on page 28)

Taking steps to avoid steps:

WordPerfect

WORDPERFECT CORP. PRODUCES WordPerfect, long considered to be one of the computer world's premier word processors. Recently, WP released an update to the Amiga version which includes many long-awaited enhancements. These changes include the addition of a typical Amiga file requester (instead of the IBM-like file requester standard with other versions of WP) and better utilization of the Amiga's multitasking abilities, offering improved editing speed when more than one window is open simultaneously. Prior to this update, WP was not the most highly-regarded Amiga word processor, but these enhancements should go a long way toward finally and emphatically changing that.

In addition to this update to WP itself, WordPerfect Corp. has released a File Conversion program and a Macro Editor, uploading both to their BBS for registered users to freely download; or, users may send \$10 to WordPerfect Corp. for a disk containing both programs. The File Conversion program converts text files between WP v1.9, WP v2.0, WP Secondary, Scribble!, and Scribble! Merge formats. The Macro Editor shows the text representation for macros in what resembles a file requester for a favorite text editor of mine (TxE+), and permits the addition or deletion of any steps (plus changes to scrolling or window size) without the need to rerecord the entire thing. This disk is a godsend to heavy WP users with both Amigas and IBMs, since it enables easy transfer of text files between different word processor formats, and helps to easily modify macros created with the powerful macro feature of WordPerfect.

An aspect of WP sadly overlooked in most Amiga magazine articles is the utilization of WP's macros. Macros are a recorded series of mouse movements or keystrokes used to perform tasks while keeping the latter to a minimum, named in one of the following ways: with an Amiga key + another key, a typed-in file name for the macro, or by pressing the 'RETURN' key instead of entering a name at the prompt (which creates a temporary macro that disappears after exiting WP). Not all keys are used in conjunction with the Amiga keys. The supported keys for macros are Amiga + : A, D, E, F, G, H, M, N, R, T, V, W, Y, and Z. WordPerfect normally utilizes the function keys F1—F10 in conjunction with the SHIFT, ALT, and CTRL keys for the word processor's operations, preventing their use as macros. Chaining macros together (where one macro calls another) is a useful feature of WordPerfect. I chained two macros together to 1) save a file after checking spelling within the document, and 2) clear the screen for a new document. Sometimes there is a need to clear a screen without saving a file, and sometimes I save a file without checking the spelling (especially on the first draft!).

The procedure used to generate these handy macros is quite simple to learn, and the tutorial aspect of WP covers it in detail sufficient for most new users. Here is a summary:

- 1) press CTRL + F10 keys;
- 2) press either left or right Amiga keys, plus another keyboard letter; at that point, WP records all mouse or function key selections—including mistakes—so plan ahead as to exactly what the macro should accomplish;

Macros

by Mike Hubbartt

- 3) make the selections the macro will perform (for example, I pressed SHIFT + F10 at this step to record a retrieve macro);
- 4) when all macro movements/selections are made, press CTRL + F10 to stop recording the macro; now press either Amiga key plus the letter that defines the macro, to activate the macro.

I use macros to customize the WP interface to personal preferences as well to generate shortcuts, although standard commands like Right_Amiga-U for enabling/disabling underlining, and Right_Amiga-B for enabling/disabling bold type are already supported by WP, and do not need a macro defined. Use either the Project menu or CTRL + F1 to open a new CLI window, or use a macro such as Amiga (either Amiga key) + n. Use the Project menu or SHIFT + F10 to retrieve or open a file for editing, or use another macro like Amiga + r. Most shortcuts I choose utilize an Amiga key + another letter because they save me time as a touch typist, and they are easier to remember than those chosen by WordPerfect.

Since macros named with Amiga key + a letter are accessed by selecting the key combination, how do you use macros with file names? To access a macro with a filename, press ALT + F10, and then type in the macro's name. Macros with file names are stored in locations other than the default directory, so they are accessed by the full path name (i.e., df1:macros/instext) whenever desired. To access a temporary macro, press ALT + F10 and then 'RETURN'. Macros can be either visible or invisible. Invisible is the default mode, which does not display the menus where selections were made in the macro definition are not shown in this mode. In

the visible macro mode, enter a delay in a macro when defining it and you can see each menu accessed by that macro as it executes.

The lack of a quick word count feature (a lifesaver for any writer) in WP is worked around via macros. The usual WP procedure to generate a word count is as follows: select the Spelling checker with either CTRL + F2—or use select SPELL from the Special menu—then select the word count option from the new window in the center of the screen. After setting up a macro, I now just need to use Amiga key + w to get the same information without performing extra steps and spending extra time. I chose Amiga key + w since the "?" sign (used in Transcript from Gold Disk) is not selectable as a macro identifier in WordPerfect.

Since the spelling of my documents is sometimes (read: often) called into question, I also have a macro (Amiga + h) to bring up the Spelling Checker with all eight options still unselected. Another favorite macro deletes text from a window, so I can start a new document (Amiga + E) instead of using F7 and answering yes or no to both exiting requesters that appear. And finally, there is a print current document macro. The normal procedure for printing out a document is: select the Full Page menu item from the Print menu, or use the SHIFT + F7 key combination and then select Full Text from the menu items, or record a macro (like Amiga + d) to save steps when sending output to the printer. My Amiga + d macro loads the printer driver information from the WP Print disk, and prints the current document without further menu selections.



•

Chaining
macros
together
(where one
macro calls
another) is a
useful
feature of
WordPerfect...

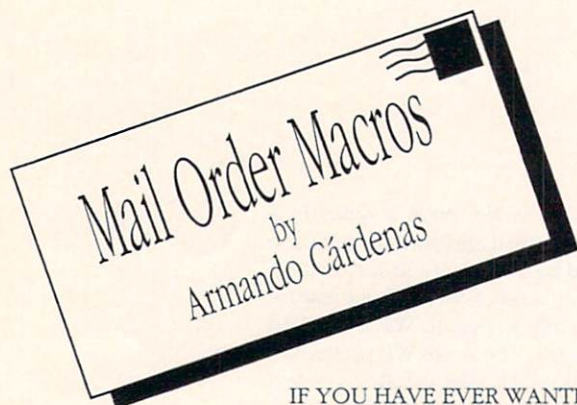
•

Here are some more helpful hints on generating macros:

Execute the sequence first, without recording the sequence, writing down all steps if the macro is quite large. In naming macros, stick with familiar labels used in past word processing applications when possible, and name newer combinations with letters fitting the description of the macro's purpose. Macros are among the best features offered in WP, and I hope all Amiga word processors eventually come to use them (at present, only one or two companies beside WordPerfect Corp. use macros in

their word processors). Owners of WP definitely need to contact WordPerfect Corp. for both the WP update and the File Conversion/Macro Editor disk, both welcome additions over the original version when writing anything from a letter to a complete book. If you still have some questions about generating macros from the method I have described here, try the examples in the WP tutorial—it's that easy to develop your own! *WordPerfect 5.1, WordPerfect Corp., 1555 N. Technology Way, Orem, UT 84057; Price: \$250.00; Inquiry #213*

•AC•



IF YOU HAVE EVER WANTED TO USE YOUR printer to address envelopes, and you have a printer with capabilities to print 1" or 1/2" from the tear-off (short tear-off), or you are willing to use printer envelopes (Memorex or equivalent)—read on.

WordPerfect and at least two other word processors include macros, which are very simple to create, and real timesavers. We can devise a macro in WordPerfect (applicable to other word processors, after doing some homework) that addresses envelopes with no extra typing required.

That is, this macro takes the page head (or your name and address) from a letter, moves it to the left, moves the name and address of the "addressee" to the center, and erases the content of the letter itself, thus eliminating the need for additional typing.

There are two requirements for this macro to be effective. The location of each line in the head of the letter (or the area from which your name and address are being taken) must be consistent, as should line format. The three or four lines containing the name and address of the addressee must always be located in the same line from the top of the form. I have a macro that types my name, address and date and ends by placing the cursor on line 10, which is the place where I always start my letters.

Name and address usually occupies 3 or 4 lines. To cover both cases, let's make the macro a 4-line macro. Type any letter using your preferred format, but place the addressee information starting at line 10. In order to make the macro work with long or short addresses, make each line about 30 characters long. These could be anything—xxxx's will do [to create the macro].

In the letter head you can include any printer commands (like double height, italics, etc.) to embellish the hard copy. These commands will transfer to the macro unless you do not want them, in

which case they can be deleted during the creation of the macro.

To start the macro, place the cursor in line 1 position 1 and define the macro as follows:

Press CTRL + F10 and at the prompt, give the Macro a simple name like AC. Press RETURN to accept the name. If you do not date your letters, skip the next step; if you do, place the cursor at the end of the line preceding the date and press RETURN 3 times. Place the cursor at the end of the last line of the address and press RETURN 3 or 4 times. This isolates the date and information to be transferred to the envelope and moves the first line of the address to line 13, the appropriate location for business envelopes.

Move the cursor to the date line and use your mouse to pull down the EDIT menu and select "Delete To End of Line". Then, move the cursor to line 1 position 1. Enter the BLOCK ON mode by depressing ALT + F4. Use the down arrow key to "BLOCK" the contents of lines 1 to 17. Enter the Move Block mode by pressing CTRL + F4 and select Cut Block 1.

Now Exit by Pressing F7, but do not save the file and do not exit WordPerfect. This erases all pages, but not the blocked rectangle. At this point the cursor is in the upper left-hand corner, which is where you want it. Press MOVE TEXT CTRL + F4 and select Retrieve Text 5.

Since printing must start at the very top of the envelope, we have to set the top and left margins; press ALT + F8, PAGE FORMAT, and select Top Margin (5), enter 0 (yes, zero top margin) and press 0 (zero) to exit. Press SHIFT + F8, LINE FORMAT, and select Margins (3), enter 2 (or whatever works best with your printer) and press 0 (zero) to exit.

To move the header to the left, press ALT + F3 to "Reveal Codes". Using the BACKSPACE key, delete the Center [C] or [A] from every line of the header. Delete the top margin normally used in your letters. Exit the "Revel Code" mode. Place the cursor at the beginning of line 13, which should be the first line of the address. Using the TAB key, move all four lines to Pos 35. Finish and save the Macro by pressing CTRL + F10 again.

To Invoke the Macro just created, load any letter, making sure the name and address starts in line 10. It can be 3 or 4 lines long. Place the cursor in line 1, press ALT + F10 and enter the name of the Macro. If you are using an Epson LQ-850 move the paper release lever back, place an envelope, and print it. If you do not have a printer with envelope-loading features, Memorex makes printer envelopes that work very nicely.

•AC•

•
Devise a
macro to
address
envelopes,
and no
extra typing
required!
•

building a RAPID FIRE JOYSTICK

THIS MONTH, WE'LL TAKE A LITTLE EXCURSION away from the parallel port. Our project this time is a rapid-fire joystick circuit for game playing on the Amiga. (although the circuit should also work equally well on the Atari, Vic-20, C-64 and C-128 computers). The circuit is placed in-between the computer and the joystick, and becomes activated via depression of the fire button on the joystick. This causes the circuit to flip-flop the voltage on pin 6 from 5 to 0 volts at approximately 17 Hz (Hz is an abbreviation for hertz which stands for "cycles per second"). This is equivalent to pressing the fire button 17 times a second. The circuit saves your trigger finger from overexertion playing those rapid fire asteroid-type games.

WHAT KIND OF GAME?

How well the circuit works depends upon how the particular game is programmed. If the game in question performs multiple firing when the fire button is depressed constantly, chances are the circuit will not improve your game score. In fact, in some cases the circuit may slow down an existing multiple-firing feature. This happens because even though the circuit oscillates pin 6 at 17 Hz, it still spends half its time with the line high (not firing). The other half of the time the line is low (firing). Depending upon the moment the computer polls the game port, it might not recognize the fire button as being depressed.

Games that require discrete (individual) fire button presses are the type of games the circuit is best suited for. The circuit flip-flops that fire pin faster than is humanly possible.

COMPUTER GAME PORTS

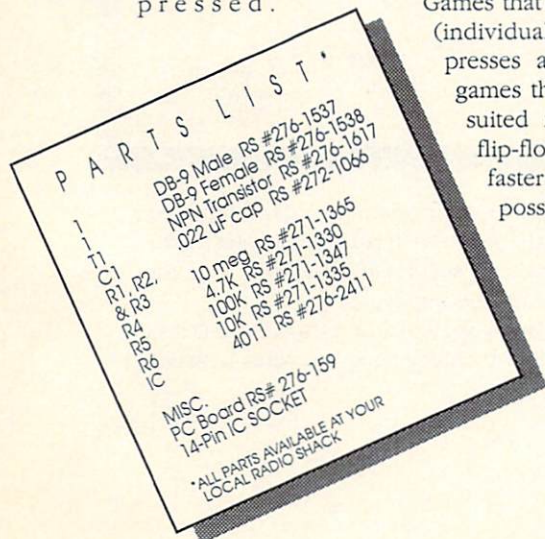
The joystick connector ports labeled Joy1 and Joy2 on the computer are standard DB-9 connectors. The DB-9 connector built into the computer is a male pin DB-9. The mating connector on the joystick is a female socket DB-9. Since we are placing our circuit in between the joystick and computer, we need both a male and female DB-9 connector on the circuit board to interface it. The female DB-9 on the circuit plugs into the game port on the computer and the joystick plugs into the male DB-9 connector on the circuit. DB-9 connectors are available from Radio Shack and most other electronic distributors.

The game ports on the Amiga are not just for joysticks, they are also used for other peripherals. In each application the designation of the pin assignment changes in respect to the peripheral in use at the time—joystick, mouse or light pen. Since this particular circuit is designed for use with a joystick, the pin assignment for the DB-9 pins is as follows:

Pin	Name	Description
1	Forward	Controller forward
2	Back	Controller back
3	Left	Controller left
4	Right	Controller right
5	Pot X	Horizontal Potentiometer (Paddle)
6	Fire	Controller fire
7	+5V	+5V power (100 ma)
8	GND	Ground
9	Pot Y	Vertical Potentiometer (Paddle)

Only three lines are utilized to interface this circuit, the +5V, GND and fire button. All of the other wires are passed through, meaning you just solder wires pin-to-pin between the male and female connectors on the circuit. See schematic Figure One. ➔

by John Iovine



It isn't necessary to remove the circuit from the computer when you are through with it. The circuit can be left on the game port. The amount of current the circuit draws from the port is minimal and should not affect normal computer operations. For most of the games the mouse can remain in Joystick port 1 for use with the workbench. Games usually utilize Joystick port 2, so there is not a conflict. But if you are using two joysticks (and circuits) for a particular game you may have to remove the circuit from Joystick port 1 when you are finished, since pin 6, when in use with a mouse, is mouse button 1. Pressing the mouse button with the circuit on will perform a rapid fire on the line, equal to 17 mouse clicks per second.

HOW JOYSTICKS WORK

The joystick is a simple device centered on 5 switches (see Figure Two). Four of the switches detect the movements of the joystick. One of the four is activated each time the joystick is moved in a specific horizontal or vertical direction. Diagonal movement is read when two adjacent switches are activated at the same time. For instance, when the joystick is pushed diagonally right, both the forward and right switches are activated. The fire button is the fifth switch on the joystick.

The lines to the joystick are active low lines. What this means is that the computer holds all the joystick lines at +5 V. When a switch is activated it connects that particular line to ground (zero volts).

4011 QUAD NAND IC

The heart of the circuit is a 4011 quad NAND integrated circuit (see Figure Three). The operation of a single NAND gate is illustrated in the truth table.

The IC (integrated circuit) in the circuit as stated is a 4011 quad NAND gate. This chip has four (quad) NAND gates. Two NAND gates can be made to oscillate using a simple RC (resistor capacitor) network (see simple oscillator Figure Three).

In our circuit the gates are made to oscillate when pin 6 (fire pin) from the joystick is brought low. The output from the IC is fed into the NPN transistor. The function of the transistor is to invert the signal output from the 4011 IC. The inverted output from the transistor is brought to pin 6 (fire pin) on the computer side of the circuit.

CONSTRUCTION

If you have been following this series and doing the projects, you should construct this circuit on the solderless bread board we have been using. Test out the circuit on the bread board before you hardwire (solder) it. This makes it easy to correct any wiring errors.

After testing, the circuit can be wired on a small piece of PC board. I used half of the PC board listed in the parts list. If you haven't done much soldering (even if you have!), it's a good idea to solder a 14 pin IC socket into the circuit instead of the actual IC. By soldering a socket into the circuit you don't have to worry about overheating and possibly burning out the IC with your soldering iron (which

Figure One:
Rapid Fire Joystick Controller

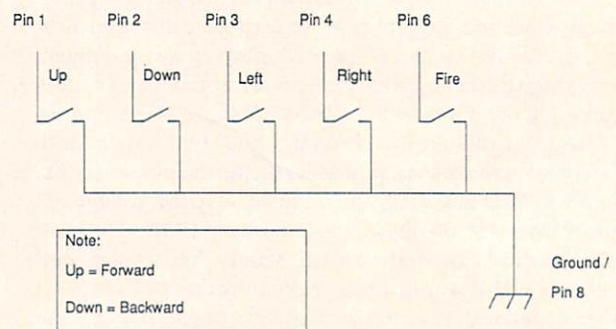
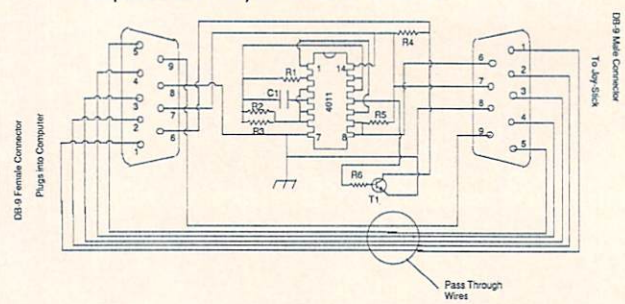
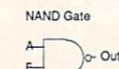


Figure Two: Joystick Switch Schematic



Truth Table

A	B	Out
L	L	H
L	H	H
H	L	H
H	H	L

Note:

L (Low) = Binary 0 or 0 V or Gnd
H (High) = Binary 1 or + V

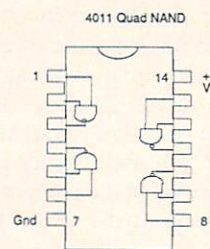
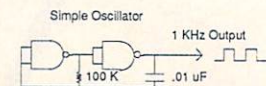


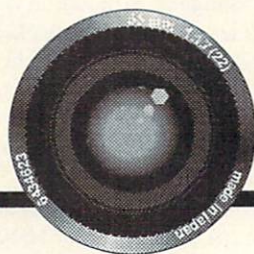
Figure Three



would surely put you in a real frustrating position). After the circuit is finished with the socket in place, simply install the 4011 IC. This also makes it simpler to replace the IC in the future, should it become necessary.

If you don't plan to use a paddle with the circuit you can simplify the circuit by eliminating the wires between pins 5 & 9.

•AC•



SNAPSHOT

by R. Bradley Andrews

LOTS OF NEW STUFF IS COMING OUT FOR THE AMIGA right now, and things look promising for future support. Hopefully, all the good titles will do very well in stores, so more companies will devote their full support to this fine machine, and we can have more of the hottest games out.

TYPHOON THOMPSON

Typhoon Thompson, a recent release from Broderbund, gets this month's award for original theme. While the far future is the setting for many games, this universe is a bit different. A tragic crash has destroyed all but one occupant of a luxury cruise ship. The remaining occupant, a small child, has landed on a water planet and has been adopted by a race of sea sprites, who do not want to relinquish their new charge.

You have been selected to redeem yourself from past mistakes by defeating several of the sprites' cities and freeing the child. Two representatives of an ancient civilization that used to occupy the planet will aid you in your quest.

The first step is to gather the four magical items needed for your final showdown with the sprites. Each magical item is held in a different sprite city. Of course, the residents will not give up their magical items without a fight. First, you must deal with each of the inhabitants of the seven islands surrounding each city by drawing them out, shooting their ships, and then scooping them out of the water into your sprite bag. Once you have gathered up all the sprites on the islands, a big sprite will trade you the magical item you seek for his captured subjects.

Battling the first city turns out to be fairly easy, as just one sprite per ship and one sprite ship per island is sent against you. But these values increase on later levels until on the final level, there are three sprites per ship and three ships per island. The top levels can get very "hairy," since the sprites must be grabbed quickly before they turn into dolphins, swim back to their island, and bring out more friends.

In addition to your jet sled and sprite-bag you have some weapons at your disposal—one added for each of the first four levels. A laser cannon is your primary weapon and is used to blast sprite ships out of the sky. The scatter bomb serves to repel all the flyers in a certain area for a while, clearing a path for your sled. The sprite magnet attracts downed sprites, very helpful when you are gathering up lots of the guys you have just shot down. Finally, if the sprite ships are on range, the freeze bomb will stop them dead in their tracks for a short time.

Sprite ships come in several varieties. Some are simply obnoxious, such as the Bumper, who thinks he is on a bumper car ride. Others may either attract or repel your ship, making for some deadly possibilities. Still others fire a variety of nasty projectiles at you, but the worst of all—the Zapper—is deadly with any contact.

The graphics in the game are done well, and very sharp. The



Broderbund's Typhoon Thompson

sound is simple, but completely adequate and also well done. The game's high point is its cartoon-like animated sequences. These begin with the intro sequence where you are knocked out and tossed into your jet sled. The sprites themselves are interesting to watch as they jump from destroyed vehicles and flounder in the water. Each possible way you can die is cleverly animated with a sequence that almost makes up for the fact that you have lost a life. Also, the sequences in which the owner of each city turns over one of the items you seek are very cute.

The mouse is used for control of the game, and this is the game's greatest shortcoming. It can be very difficult to maneuver in three dimensions. While it may be nice to have two different buttons and more firing options available right on the mouse, I still would have preferred some sort of joystick interface augmented with some key presses. It is too bad someone doesn't come out with a good two-button mouse. The Atari-compatible joystick is just too entrenched.

There is another problem related to the interface. While you are supposed to be able to easily move near a downed sprite and

stuff it into your bag, doing so takes a lot of practice, and it can be very difficult to actually fly the right pattern to grab the critter. You must also move at least a bit between each sprite grab. I often found myself in the midst of a whole bunch of sprites, able to grab only one or two before I had to defend against an incoming ship. This problem is compounded at the higher levels, given that there are more sprites per ship. It is vital to grab all the sprites from each ship, or they will go back and form another ship full of pesky sprites to harass you.

In spite of this, I actually had a good time with the game. I did manage to hang the game with a "too many objects" message trying to rescue the child at the fifth and final level, because I found it very difficult to identify, let alone grab the three sprites that belonged to any one ship. But, I do think that with a little practice, I could even manage to beat that level. At one point on the next-to-last level, sprites were multiplying like rabbits, but I learned to take out one ship at a time, and everything was OK.

While it may not be worth a lot due to the limited number of levels and the frustrating interface, I did enjoy Typhoon Thompson, and recommend you check it out at the store.

DAY OF THE VIPER

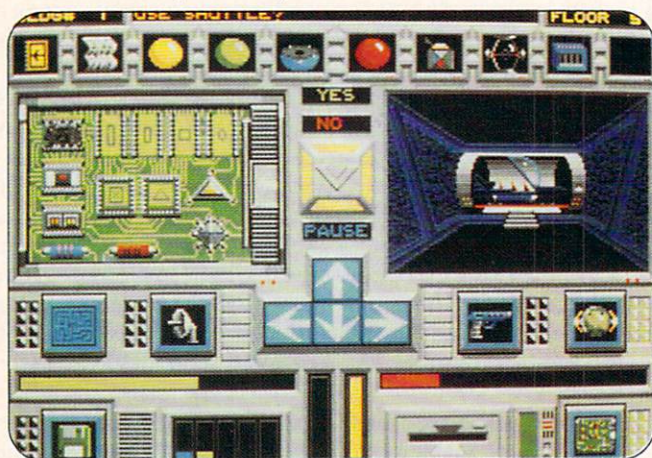
Day of the Viper from Accolade is also set far in the future, but this time it is a rogue machine that threatens mankind. Several sentient androids were created to aid man in the hazardous tasks that normally accompany space exploration. Unfortunately, something caused them to go crazy after only a short time. Now, led by the devious GAR, they are headed back to earth with destruction in mind.

The only thing that can stop them is the sole operational Viper remote-controlled combat vehicle. Because of its size, it can sneak into their main base undetected. From there it can gather data programs needed to activate the auto-destruct sequence on the base's computer, thereby destroying the menace.

The base consists of five different complexes with five levels each. Your adventure will take you through these levels as you look for the keys to GAR's destruction. Scattered about the complex are useful items, such as keycards that open colored doors to your vehicle, energy crystals that can be used to refill your internal batteries, repair modules to fix damage to your craft, and more.

While you may slip in undetected, the base remains hostile to your presence. Enemy robots patrol the halls, dancing gingerly in front of you until you destroy them. Mines and other traps also block key passageways and must be either bypassed or detonated.

Accolade's Day of the Viper



The real key to your mission lies in the 25 data diskettes that contain the destruct code. One is somewhere on each level, and only by inputting these into the base computer can you complete your task; everything else is really just a step along the way.

The center-right portion of the screen features a three-dimensional view common to many dungeon exploration games, while the center-left portion is used as a modifiable display, the primary functions of which are to show a map of the current level and the list of messages you have stored on your electronic notepad. Your current inventory of items is displayed at the top of the screen, and the bottom of the screen holds various action buttons that come into use during the game.

The graphics look nice and add to the feel of an underground complex. And the enemy robots almost literally dance about the screen in front of you, often a frustrating occurrence. The sound is simple, but complements play. The mouse is used for all control during the game and works fairly well, though having keyboard equivalents would have been nice.

One thing I should note: this game is VERY similar to the game Slaygon, released a while back by Michtron. The graphics are a little better, and the explorable map is much larger (Slaygon only had one level as I recall), but most of the other elements are virtually identical. I wonder if Michtron knows about the earlier game?

This game is very frustrating. Enemy robots are a little too fast, and it is very easy to die quickly, as they blast at you before you

can use properly placed shots to destroy them. The setup of the base in each level is random. While the rooms and walls are always in the same location, the objects (good and bad) are often scattered in different locations. While this makes for some variety, it can be frustrating if a needed repair module is hidden behind a door your keycard cannot open.

Yes, the game is a challenge. There is enough depth to allow for a lot of playing time. Unfortunately, after a little while, play just becomes maddening. But I did manage to do a little better each time I played, though I finally gave it up after only getting through most of the first building. I just could not find any new keycards or sufficient repair modules, so death was pretty rapid at the deeper levels. The game is worth considering if you like exploring and are good at quick reaction using the mouse.

WEIRD DREAMS

Weird Dreams also wins an award for being bizarre, as you take a journey into the mind of a hospital patient undergoing

Microplay Software's Weird Dreams



Order Toll Free 800-558-0003 *Since 1982* **ComputAbility** *Consumer Electronics* Order Toll Free 800-558-0003

AMZ 6-90

MASTER 3A Disk Drive \$115*

FREE DELIVERY to the 48 Contiguous States

IVS Trumpcard Hard Drive Packages for A2000 Series

Seagate	
ST-125N 20 MEG.....	399 DEL
ST-138N 30 MEG.....	499 DEL
ST-157N-1 49 MEG-28ms.....	479 DEL
ST-177N 60 MEG NEW I.....	539 DEL
ST-277N-1 60 MEG.....	599 DEL
ST-296N 80 MEG.....	539 DEL

Quantum Pro Drive	
40 MEG SCSI Pro Drive.....	529 DEL
80 MEG SCSI Pro Drive.....	749 DEL
100 MEG SCSI Pro Drive.....	839 DEL

these kits include IVS Trumpcard SCSI hard drive ctrl, cable, software and FREE delivery in the contiguous USA. This is not an assembly kit! It is a package.

Hard Drive Cards (A-2000)

Seagate	
ST-125N 20 MEG.....	409 DEL
ST-138N 30 MEG.....	429 DEL
ST-157N-1 49 MEG-28 ms.....	499 DEL
ST-177N 60 MEG NEW I.....	559 DEL

Quantum Pro Drive	
40 MEG SCSI Pro Drive.....	549 DEL
80 MEG SCSI Pro Drive.....	789 DEL
100 MEG SCSI Pro Drive.....	859 DEL

these cards include IVS Trumpcard ctrl, mounting brkt, cbl, software, and FREE delivery in the contiguous USA.

PRICES SLASHED SUPRA MEMORY

Supra Ram 500.....	79 Del
Supra Ram 2000-2 MB.....	289 Del
Supra Ram 2000-4 MB.....	449 Del
Supra Ram 2000-6 MB.....	609 Del
Supra Ram 2000-8 MB.....	769 Del

FREE DELIVERY In the Contiguous 48 States

CALL FOR ALL YOUR AMIGA™ NEEDS

Trumpcard 500 Hard Drive Packages for Amiga 500

Seagate	
ST-125N 20 Mb.....	459 Delivered
ST-138N 30 Mb.....	479 Delivered
ST-157N-1 49 Mb-28ms.....	549 Delivered
ST-177N-60 Mb NEW.....	609 Delivered

Quantum Pro Drive	
40 MEG SCSI Pro Drive.....	599 DELIVERED
80 MEG SCSI Pro Drive.....	839 DELIVERED
100 MEG SCSI Pro Drive.....	909 DELIVERED

These Packages include Trumpcard 500 enclosure, IVS Trumpcard SCSI hard drive ctrl, software & free delivery in the contiguous USA.

Trumpcard 500 Ram Expander available - 512k • 1 MEG • 2 MEG • 4 MEG

PRINTERS

Panasonic	
KX-P1124	\$277

Panasonic 1180.....	179
Panasonic 1191.....	239

star

NX1000	NX1000
Multifont 2	Rainbow
\$158	\$205

StarXB 2410.....415

Quantum Pro Drives

40 MEG SCSI Pro Drive.....	369
80 MEG SCSI Pro Drive.....	599
100 MEG SCSI Pro Drive.....	679

NEW XETEC Fast Card + Hard Drive Packages for Amiga 2000

Seagate	
ST-125N 20 MEG.....	439 Del
ST-138N 30 MEG.....	459 Del
ST-157N-1 49 MEG-28ms.....	529 Del
ST-177N 60 MEG NEW I.....	589 Del
ST-277N-1 60 MEG.....	559 Del
ST-296N 80 MEG.....	589 Del

Quantum Pro Drive	
40 MEG SCSI Pro Drive.....	559 DEL
80 MEG SCSI Pro Drive.....	799 DEL
100 MEG SCSI Pro Drive.....	879 DEL

Card features Xetec's DMA hard, outboard ROM, auto-config circuitry, 25 pin SCSI core, disk, utilities manual.

FREE DELIVERY IN THE CONTIGUOUS 48 STATES

XETEC Fast Card System Hard Drive Packages for Amiga 500

Seagate	
ST-125N 20 MEG.....	599 Del
ST-138N 30 MEG.....	619 Del
ST-157N-1 49 MEG-28ms.....	689 Del
ST-177N 60 MEG NEW I.....	749 Del
ST-277N-1 60 MEG.....	719 Del
ST-296N 80 MEG.....	749 Del

Quantum Pro Drive	
40 MEG SCSI Pro Drive.....	739 DEL
80 MEG SCSI Pro Drive.....	979 DEL
100 MEG SCSI Pro Drive.....	1049 DEL

System includes Adaptor with autoboot ROM, enclosure manual, Fast Track disk with utilities & shielded SCSI cable. With power supply and fan.

FREE DELIVERY IN THE CONTIGUOUS 48 STATES

Seagate SUPER SALE

ST-125N 20 MEG - SCSI.....	235
ST-138N 30 MEG - SCSI.....	255
ST-157N 49 MEG - SCSI.....	295
ST-177N-1 49 MEG - 28 ms.....	319
ST-177N 60 MEG - SCSI.....	379
ST-225N 20 MEG - SCSI.....	255
ST-277N-1 60 MEG - SCSI.....	345
ST-296N 80 MEG - SCSI.....	379
ST-1096N 80 MEG - 24ms.....	435

VIDEO PACKAGE

PANASONIC 1410 CAMERA
16MM LENS WITH
VARIABLE IRIS
COPYSTAND WITH LIGHTS
DIGIVIEW GOLD 4.0
\$419 DELIVERED!

GVP LOWER PRICES

HC/O SCSI Host Adaptor.....	199
HC8/O-HD Contr-O/K exp. 8 MB.....	229
GVP 68030/28 4MB w/Math.....	1999 DEL

GEN LOCKS

Ami Gen.....	95
Magni 4004 w/ Remote.....	1569
Scan Lock.....	739
Super Gen.....	659
Super Gen SC.....	CALL
Super Gen 2000S.....	1819 DEL

MISCELLANEOUS HARDWARE

Flicker Fixer.....	\$429
Perfect Sound.....	\$69
Color Splitter.....	\$119
Migraph Hand Scanner.....	\$319
Sharp JX 100 Color Scanner.....	\$759
Cordless Mouse.....	\$99
GI-500 OptoMechanical Mouse.....	\$49
GI-1000 Fully Optical Mouse.....	\$85
Gravis Mouse Stick.....	\$85
Supra 2400zi Internal (A2000).....	\$139
Baud Bandit Modem.....	\$105
Mega Midget Racer 25 mg.....	\$669
Supercard.....	\$75
Action Replay.....	CALL
Syncho Express.....	\$55

Supra 2400

Ext. Modem
& Cable
\$115
FREE DELIVERY to 48 Contiguous States

688 Attack Sub.....	31.95	Chaos.....	24.95	Empire.....	31.95	Laser Squad.....	24.95	Persian Gulf Inferno.....	24.95	Shadow of the Beast.....	31.95	Thinker.....	57.95
A-Talk 3.....	64.95	Chessmaster 2100.....	31.95	Excellence 1 Meg.....	158.95	Last Duel.....	24.95	Phasar 4.0.....	24.95	Shark Attack.....	24.95	Third Courier.....	30.95
Advantage, The.....	114.95	Chronoquest II.....	30.95	Eye of Horus.....	23.95	Leisure Suit Larry.....	25.95	Photo Paint 2.0.....	87.95	Shufflepuck Cafe.....	24.95	Thunder Blade.....	29.95
After Burner.....	30.95	Clue.....	24.95	F-18 Combat Pilot.....	31.95	Leisure Suit Larry II or III.....	37.95	Pic Magic.....	57.95	Slm City.....	30.95	Times of Lore.....	24.95
Albion Ranger.....	28.95	Codemaster: Iseman.....	37.95	F-19 Stealth Fighter.....	CALL	Life and Death.....	31.95	Pratesi.....	28.95	Terrain Ed.....	14.95	Trackers Quest.....	21.95
All Dogs Go To Heaven.....	28.95	Colonel's Bequest.....	37.95	F/A18 Interceptor.....	31.95	List & Label.....	30.95	Planet of Lust.....	24.95	Skate or Die.....	25.95	Treasure Trap.....	25.95
Altered Beast.....	30.95	Comic Setter.....	59.95	Falcon.....	31.95	Logiworks.....	184.95	Police Quest I.....	30.95	Smooth Talker.....	25.95	Trumpcard Disk Mgr.....	89.95
Armagadon.....	87.95	Comic Art Disks (each).....	20.95	Operation Counterstrike.....	16.95	Loom.....	37.95	Police Quest II.....	37.95	Softcore Royal.....	19.95	Tunnels of Armageddon.....	25.95
Aquanaut.....	25.95	Conquest of Camelot.....	37.95	Fat Tracks.....	34.95	Lords of the Rising Sun.....	31.95	Pool of Radiance.....	31.95	Space Ace.....	34.95	Turbo Out Run.....	31.95
Archipelago.....	23.95	Creatures.....	37.95	Federation.....	31.95	Lords of War.....	24.95	Populous.....	32.95	Space Harrier.....	30.95	Turbo Silver 3.0 1 Meg.....	112.95
Audio Master II.....	62.95	Cribbage King/Gin King.....	25.95	Flight Simulator II.....	30.95	M2 Amiga.....	147.95	Power Windows 2.5.....	54.95	Space Quest 1 or 2.....	30.95	TV Text.....	64.95
Archie C Developer V5.0.....	177.95	Cross Doss.....	18.95	Foot's Errand.....	31.95	M2 Amiga Debugger.....	77.95	Powermover.....	25.95	Space Quest 3.....	37.95	TV Text Professional.....	99.95
Archie C Pro V5.0.....	119.95	Curse of the Azure Bonds.....	26.95	Future Wars.....	30.95	Manhattan - NY or SF.....	31.95	Print Master Plus.....	24.95	Space Rogue.....	30.95	Typhoon of Steel.....	31.95
B.A.D.....	29.95	Cycles.....	30.95	Gauntlet II.....	30.95	Mantra.....	28.95	Pro Motion.....	51.95	Star Flight.....	31.95	Typhoon Thompson.....	30.95
B.A.T.....	31.95	DB Man.....	167.95	Genghis Khan.....	41.95	Mania Marston.....	28.95	Pro Tennis Tour.....	25.95	Star Trek: The Final Front.....	34.95	Ultima II.....	24.95
Balance of Power 1990.....	30.95	De la Viper.....	30.95	Grand Prix Circuit.....	27.95	Matrix Raiders.....	21.95	Pro Video Gold.....	139.95	Stellar Crusade.....	36.95	Ultima III.....	38.95
Bar Games.....	28.95	Death Bringer.....	25.95	Gunship.....	34.95	Maxi Beacon Typing.....	31.95	Fort Packs.....	59.95	Storm Across Europe.....	37.95	Ultima V.....	38.95
Barney Bear Goes to School.....	21.95	Deluxe Music Constr 2.0.....	61.95	Halls of Montezuma.....	28.95	Maxipix 3.....	147.95	Pro Write 3.0.....	94.95	Storm Rod.....	31.95	Ultra Design.....	219.95
Baron of the Rings.....	24.95	Deluxe Paint III.....	94.95	Hardball II.....	30.95	Micro Fide Filter Plus.....	114.95	Pro Fonts 1 or 2.....	21.95	Strike Ace.....	30.95	Ultracard Plus.....	57.95
Bards Tale II.....	38.95	Deluxe PhotoLab.....	94.95	Heat Wave.....	30.95	Microleague Wrestling.....	24.95	Professional Draw.....	112.95	Strip Poker II.....	23.95	Universe 3.....	31.95
Batman.....	28.95	Deluxe Print II.....	51.95	Heroes Quest.....	37.95	Might Rec Studio V1.1.....	39.95	Professional Page.....	167.95	Styx.....	21.95	Vette.....	30.95
Barney Bear Goes to School.....	21.95	Deluxe Productions.....	128.95	Hillstar.....	31.95	Might and Magic II.....	37.95	Project D+ Backup.....	30.95	Stunt Track.....	21.95	Video Effects 3-D.....	113.95
Battle Chess.....	29.95	Deluxe Video III.....	94.95	Hound of Shadow.....	25.95	Modeler 3D.....	62.95	Prospector Mazes of Zor.....	24.95	Super Contra.....	28.95	Video Tiler V1.5.....	92.95
Battle Hawks 1942.....	31.95	Designasaurus.....	30.95	Haynes Book of Games.....	21.95	Monopoly.....	24.95	Puffy's Saga.....	25.95	Super Star Basketball.....	CALL	Videoscope 3D.....	122.95
Battle Squadron.....	24.95	Digipaint 3.0.....	64.95	If It Moves Shoot III.....	19.95	Movie Setter.....	59.95	Puzzle Storybook.....	25.95	Superbase Personal.....	46.95	Viva-Amiga.....	122.95
Battles of Napoleon.....	31.95	Digipaint 3.0.....	64.95	Imperium.....	25.95	Music X.....	169.95	Quarter Back.....	42.95	Superbase Personal 2.....	87.95	Voice Recognition.....	109.95
BattleTech: The Crk Inc.....	30.95	Digiview Gold.....	124.95	Indiana Jones-Action Game.....	25.95	Neuroancer.....	28.95	Rambo III.....	22.95	Superbase Pro 3.0.....	189.95	Vortex.....	24.95
Baud Bandit.....	30.95	Dinosaur Discovery Kit.....	25.95	Indiana Jones-Graphic Adv.....	30.95	New York Warriors 1 Meg.....	30.95	Rampage.....	24.95	Superbike Simulator.....	30.95	War in Middle Earth.....	30.95
Bermuda Project.....	24.95	Disk Master.....	28.95	Infestation.....	24.95	Night Force.....	28.95	Raw Copy.....	34.95	Superplan.....	25.95	War of the Lance.....	25.95
Beyond Dark Castle.....	22.95	Distant Sun.....	40.95	Introcad.....	49.95	Night Hunter.....	25.95	Red Storm Rising.....	34.95	Swords of Aragon.....	25.95	Waterloo.....	37.95
Blade of Steel Hockey.....	28.95	Dos 2 Dos.....	30.95	Invasion.....	174.95	Nuclear War.....	30.95	Renegade.....	22.95	Swords of Twilight.....	31.95	Wayne Gretzky Hockey.....	30.95
Block Out.....	25.95	Double Dragon II.....	25.95	Iron Lord.....	25.95	Omega.....	30.95	Rhyming Notebook.....	25.95	T.V. Show V2.0.....	64.95	Wild Dreams.....	24.95
Blue Angels.....	30.95	Dragon Force.....	31.95	Island of Lost Hope.....	23.95	Omniplay Baseball: 5 on 5.....	30.95	Rick Dangerous.....	21.95	T.V. Sports Basketball.....	31.95	William Tell.....	25.95
Bolix - The Game.....	24.95	Dragon's Lair.....	33.95	It Came From The Desert.....	31.95	Omniplay Horse Racing.....	30.95	Rick Davis Soccer 1 Meg.....	30.95	T.V. Sports Football.....	31.95	Wind Walker.....	25.95
Brain Blaster.....	25.95	Dragon's Lair II.....	43.95	Jack Nicklaus Golf.....	30.95	On Line Platinum.....	58.95	Rings of Medusa.....	28.95	Targhan.....	24.95	Wings.....	31.95
Broadcast Tiler.....	CALL	Joan of Arc.....	28.95	Joan of Arc.....	28.95	Over Run.....	37.95	Risk.....	24.95	Tax Break 2.0.....	49.95	Wings of Fury.....	24.95
Buddkan.....	25.95	Kampfgruppe.....	37.95	Kampfgruppe.....	37.95	Operation Wolf.....	22.95	Robocop.....	28.95	Teenage Mutant Turtles.....	22.95	Worlds Platinum, The.....	167.95
Can Do.....	84.95	Karafonts Headlines 1 or 2.....	44.95	Karafonts Headlines 1 or 2.....	44.95	Page Renderer 3-D.....	94.95	Rocket Ranger.....	31.95	Tetris.....	20.95	Works Atlas.....	35.95
Carmen Sandiego/Europe.....	30.95	Keybd. Control Seq L3.....	212.95	Keybd. Control Seq L3.....	212.95	Page Setter 2.....	74.95	Romance/Three Kingdoms.....	44.95	Test Drive 2 - The Duel.....	30.95	World Class Leader Bd.....	28.95
Carmen Sandiego/USA.....	30.95	King's Quest 1, 2, or 3.....	30.95	King's Quest 1, 2, or 3.....	30.95	Scene Generator.....	24.95	Scene Generator.....	24.95	Cal Challenge.....	24.95	Xenophobe.....	25.95
Carmen Sandiego/World.....	30.95	Knights of Legend.....	37.95	Knights of Legend.....	37.95	Scrabble.....	24.95	Scrabble.....	24.95	European Challenge.....	15.95	Zak McKracken.....	28.95
Cham/Sci-Mut Priestess.....	30.95	Kristal, The.....	31.95	Kristal, The.....	31.95	Scribble Platinum.....	87.95	Sex Vixens-Outer Space.....	22.95	Muscle or Super Cars.....	15.95	Zork Zero.....	38.95
Champions of Krynn.....	31.95	Pen Pal.....	84.95	Pen Pal.....	84.95	Sex Vixens-Outer Space.....	22.95	Their Finest Hour.....	37.95				

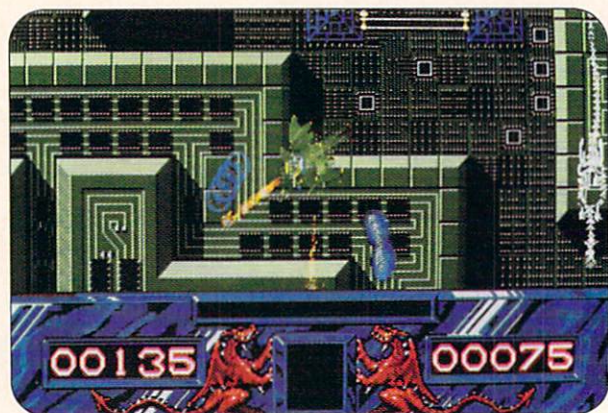
MasterCard VISA

NO CREDIT CARD SURCHARGE

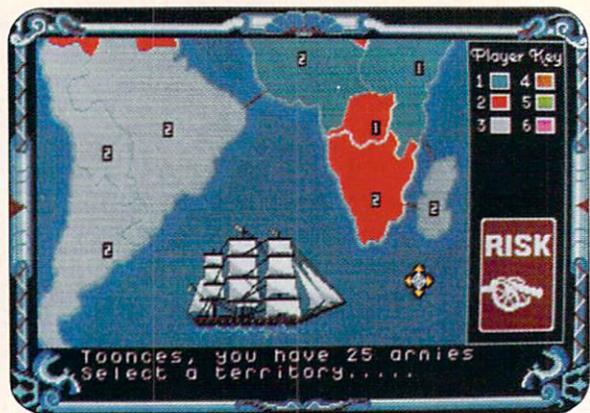
HOURS MON-FRI 9am-9pm SAT 11am-5pm P.O. BOX 17882 Milwaukee, WI 53217

INFORMATION 414-357-8181 FAX 414-357-7814

ORDERING INFORMATION: Specify system. For fast delivery send cashier's check or money order. Personal and company checks allow 14 business days to clear. School P.O.'s welcome. C.O.D. charges are \$4.00. In Continental U.S.A. include \$4.00 for software orders 5% shipping for hardware, minimum \$5.00. MasterCard and Visa orders please include card #, expiration date and signature. WI residents please include 5% sales tax. HI, AK, FPO, APO, Puerto Rico and Canadian orders, please add 8% shipping, minimum \$8.00. All other foreign orders add 15% shipping, minimum \$15.00. All orders shipped outside the Continental U.S.A. are shipped first class insured U.S. mail. If foreign shipping charges exceed the minimum amount you will be charged the additional amount. All goods are new and include factory warranty. We do not guarantee compatibility & version fit. Due to our low prices all sales are final. All defective returns must have a return authorization number. Please call (414) 357-8181 to obtain an R.A. # or your return will not be accepted. Prices and availability subject to change without notice. Shipping and handling are non-refundable. We ship the latest versions available to us, updates must be handled by and user directly with the manufacturer.



Free Spirit's Dragon Scape



Virgin Mastertronic's Risk

surgery, and obviously under the influence of some pretty serious anesthesia!

The subconscious, appropriately enough, is a weird place. Giant bees just want to sting you for the fun of it, two-legged gargoyles want to stomp you, and nasty flowers want to eat you for lunch. Not a nice place at all ... maybe it was that pizza you had last night for dinner?

Fortunately, there are many items in this realm you can use to defeat the obstacles set before you. A stick can beat the flowers into the ground and a fly swatter can scare the bee back. The other obstacles can also be overcome, but it does take a bit of creative thinking to do so.

The graphics in the game are very sharp and clear. They obviously took some time here. The backgrounds are full of rich detail and the characters move very smoothly across the screen. Hopefully, Microprose will realize how good this looks, and make their other games just as sharp.

Control is via the joystick, which generally works well, but can also prove to be a major sticking point. While the puzzles do take a while to figure out, implementing the solutions often causes problems, mostly due to delays by your onscreen alter ego in executing your commands. Because there is no save option, each game involves working through the same basic and repetitive puzzles as you seek to make it to the later ones with sufficient strength to confront your adversaries.

In spite of this, a complete play-through of the game is rather short, once you get all the tricks down. Perhaps this is why they don't offer a save feature, with that the shortness would have really been apparent.

Weird Dreams is a bit too bizarre for my tastes and the control is too difficult for me. But those who really enjoy solving puzzles and who are moderately good with a joystick may have a good time.

DRAGON SCAPE

Dragon Scape is a recent arcade release from Free Spirit Software. As usual, the kingdom is lost and it is up to you to restore it. Only by gathering each of the magical relics and restoring them to their proper place can you save your planet.

Each level in Dragon Scape has eight different pairs of relics that must be matched up. While you are riding a dragon, the objects themselves are mostly contemporary in nature: a drill bit must be dropped on a drill, a cassette tape must be reunited with its portable player, and so on. I am not sure how many levels the game has in total since the rules are a bit sparse and I was only able to get into the second level after several hours of playing.

As your energy drains from flying around and making contact with aliens, your dragon loses its skin and becomes just a skeleton. If the skin vanishes completely, you explode and the game is over. A red power pill is present in most levels and can be used to recharge your dragon to full strength by direct contact. It doesn't really want to be used, though, so you must trick it into letting you touch it.

The graphics are once again very nice and up to arcade quality. The background scrolls very smoothly and many objects can be on the screen at once without noticeable slowdown. The sounds are digitized, producing a smooth "You got it" and "Yeah!" when you grab and place an artifact correctly.

The problem with this game is mostly one of control. The dragon is slow to respond to your commands, and often takes a while to actually go in direction you aim him. And the re-power dot seems to hang out right in the middle of a bunch of enemies. For the most part they can be ignored, but they do drain some strength. And shooting them seems rather worthless since they just seem to quickly regenerate, while you tend to take damage as you fly into your own flames. Generally I just avoided firing and making contact.

While the game has fabulous graphics and nice sound I cannot recommend it because of the extreme difficulty of successful play. Now if they use what they have learned here and put a good interface on top of it, we should have a product people will line up to grab.

DOUBLE DRAGON II

Double Dragon II follows the themes begun in the original Double Dragon. One sole survivor of the original enemy gang has somehow revived the gang leader and his cronies. Naturally, his first act was to kidnap and torture your girlfriend. Once again, it is off through waves of bad guys with only your fast wits and powerful body to aid you.

The bad guys come in a wide variety of shapes and sizes. Some are relative pushovers and others will take quite a beating before they go down for good. Some even fight with deadly weapons that you can capture and use for the remainder of the current level once you defeat them.

Not one game discussed here has bad graphics, and Double Dragon II is no exception. The backgrounds are detailed and scroll smoothly while the characters move easily as well. The sounds are nice and the joystick control is very easy to use.

The only major control problem I had was when I got knocked down and then had to get up in the midst of a group of

bad guys. Usually they would get a shot in and I would be down on my back repeatedly until I finally lost my present life.

Other than that I didn't have too much trouble doing in my opponents. But that is the other problem. The game seemed really pointless. It was just a series of successful jump kicks and marches along the way, until I got caught in the midst of a group of bad guys. Then I would tend to die quickly, but restart with a new life and mow them down from the outside. My game finally ended when I met up with a sword-swinging samurai who was almost untouchable. After a few minutes I was out of lives (he was barely scathed) and the game was over. This man serves as a guard over the rest of the game, preventing any future progress.

The game is actually OK as far as arcade action goes. Perhaps I am just getting jaded with the genre and want more thinking involved. While I like some mindless entertainment, this seemed a bit too mindless for me. And then hitting that brick (or sword) wall was frustrating to no end. Still, the game is sure to please the true action fan.

RISK

Adding to their line of Parker Brother's licenses, Virgin Mastertronic has made Risk available to Amiga owners everywhere. While most people have not heard of some of the other wargames discussed in this column, nearly everyone has heard of Risk. With its intelligent mix of a little bit of strategy and a lot of luck, the game has become a favorite of many game players, young and old.

For those who have lived in a cave for most of their life, the basic goal in Risk is to conquer the world. The map shows the entire earth from an overhead perspective, divided up into 42 different territories. The person who winds up occupying all of them wins.

No navy or air force here—armies hold sway, and are the sole implement of foreign policy. Each turn, a player gets a number of new armies based on how many territories he controls, plus bonus armies in the event he controls any of the five major continents. The player adds these new forces to existing ones, and then attacks any adjacent territory he chooses. To end his turn, the player takes one "free" troop movement to consolidate or solidify his global forces, and draws a risk card, only if he was successful in capturing at least one enemy territory.

The risk cards are a focal point of play, since they can be turned in for increasing numbers of bonus armies as the game progresses, encouraging broader invasions and expansion attempts.

The dice are the sole determiner of combat. The attacker rolls up to three, one per attacking army (he can attack with more than three armies, but can only roll a maximum of three dice), while the defender is limited to a maximum of two (again, one per defending army).

The dice are ranked in order and matched to see who loses armies, with the defender winning all ties. Sheer odds often work, but it is not surprising to have a sole army in Alaska fight off a massive invasion from Asia, thus saving the Americas from domination.

Two to six players can compete in the Amiga version of Risk, with the computer filling in for any or all positions. Four different variants are covered: The U.S. Full Game, the U.S. Short Game, the U.K. Full Game, and the U.K. Short Game.

The mouse is used for all control during play and works fairly well, though it can be a bit tedious for some actions. Keyboard equivalents are available for some functions, helping out here.

The graphics are nice—but not spectacular—and the sound for the game is limited to simple beeps when actions are carried out.

The map scrolls around and only about a sixth of the world is visible at a time.

The game is a bit slow with computer players. They can take forever to make up their minds, and then their moves are often pretty poor. Even when they get on a roll, they tend to make the mistake of overextending themselves and become open to easy conquest, because their forces are spread too thin.

While the game is enjoyable, Risk for the Amiga falls short of utilizing the computer's full potential. Other more detailed conflict games are out, or are coming out soon. I would recommend skipping this one, unless you just love the board game.

Remember to take a break to see the sunshine during all your gaming this summer!

•AC•

THIS MONTH'S GAMES

Typhoon Thompson, Search for the Sea Child
Broderbund Software Inc.

17 Paul Drive
San Rafael, CA 94903
(800)521-6263
Price: \$44.95
Inquire #207

Day of the Viper
Accolade

550 South Winchester Boulevard, Suite 200
San Jose, CA 95128
Price: \$19.95
Inquire #208

Weird Dreams
Microplay Software
180 Lakefront Drive
Hunt Valley, MD 21030
(301)771-1151
Price: \$39.95
Inquire #209

Dragon Scape
Free Spirit Software
P.O. Box 128
58 Noble Street
Kutztown, PA 19530
(215)683-5609
Price: \$39.95
Inquire #210

Double Dragon II
Virgin/Mastertronic
18001 Cowan, Suites A & B
Irvine, CA 92714
(714)833-8710
Price: \$39.99
Inquire #211

Risk
Virgin Mastertronic
18001 Cowan, Suites A & B
Irvine, CA 92714
(714)833-8710
Price: \$39.99
Inquire #212

KEYCRAFT

regularly
\$49.95
now only
\$29.95

Become a power user with KeyCraft, the keyboard and mouse macro-maker for the Amiga. Customize applications to work the way you want them to... Double your productivity... Triple your typing speed... Create realtime animations with any paint program! Act now to receive KeyCraft at this low introductory price (don't forget to ask for your free "PD-Power")

version 1.3
Pro POD
regularly \$39.95
now only \$29.95
Online Database
C and assembler

StraightLine Software Inc.

The Shortest Distance Between Two Points
3818 North Kimball Avenue • Chicago, IL 60618
Mail orders to: SSI • Box 59800 • Chicago, IL 60659
When ordering by mail, include \$2 s & h. Illinois residents,
please add 7% sales tax.

Stop searching through those RKMs! ProPOD provides instant access to all system structures and function-call templates at the touch of a key! You'll save valuable programming time when you let ProPOD "look it up" for you! Guaranteed to increase your coding productivity, or your money back!

To order, call 1-800-458-2995
IL & overseas call 1-312-539-5506
Visa, MasterCard, and CODs accepted.
Be sure to ask for your free "PD-Power" disk.
Introductory prices valid until 2/15/90.

Circle 106 on Reader Service card.

Free Software!

Be one of the first 100 people to purchase KeyCraft or ProPOD from this ad and we'll include "PD-Power", our exclusive collection of power utilities for the Amiga, **FREE!**

Win \$2000 in Fish!

We're giving away the entire Fred Fish Collection (estimated \$2000 value) to one lucky member of the Amiga community. To enter, send a self-addressed, stamped envelope to:

SSI Free Fish Contest
PO Box 59800
Chicago, IL 60659

One entry per person. No purchase necessary. Void where prohibited. Winner's name will be published in the June, 1990 Amazing Computing. Contest ends 4/15/90, and entries must be received by then. Rules will be sent to all contestants (that's why the SASE). **Good Luck!**

(PageStream, continued from page 17)

Since Postscript is available in the QMS PS-810 plus, there is no need for me to run it in HP emulation mode anyway. I bring it up only because there might be other LaserJet-compatible printers that won't work properly with the program. If you own PageStream and have decided to take the plunge and buy a laser printer, be sure you have the option to return it if PageStream doesn't drive it properly.

As is the case with most productivity software, the PageStream manual itself is not very user-friendly, in that the index is entirely inadequate. Every time I tried to find a specific entry it was not in the index, and I ended up paging through the manual looking for what I wanted. For example, there were several items I wanted to find out about while working with Page values. The only entry in the index that starts with Page is Page Numbering. How about Page Creation, Page Deletion, Page Moving, or any number of permutations involving the page? Indeed, the entire index for this sophisticated desktop publishing package manual is less than two 5 1/2 x 8 1/2 inch pages. There is also no entry in the index for Technical Support or Customer Support. Appendix A is titled Customer Support and contains two pages of technical support information such as phone numbers and other pertinent information about Soft-Logik's support policies. I could not find a technical support telephone number and spent much time

trying to reach their sales office to get the number. In fact, none of the appendices are listed in the Table of Contents. I did find a "Quick Contents" page in the manual (immediately following the cover page) that does list the Appendix contents, but it is located ahead of the front matter, not where I would normally expect to find it.

My last major complaint is with regard to the list of supported import filters. PageStream is lacking in structured drawing import filters. The only two formats supported are Aegis Draw and MetaFile formats. I don't know of any other program that supports the MetaFile format, though Aegis Draw is a rather popular structured drawing format on the Amiga. There should also be an HPGL format import as many drawing programs have drivers that can generate plots on the Hewlett-Packard Graphics Language plotters. The import filter that is really missed by this desktop publisher is one written to import files created from Professional Draw. Professional Draw is currently the only professional grade illustrator program available for the Amiga—its lack of support in PageStream is almost inexcusable. Soft-Logik's answer to this is that Gold Disk, the publisher of Professional Draw, won't release details of the file format, and is keeping the format information a proprietary secret.

PageStream does import Encapsulated Postscript files, and Professional

Draw does create the EPS format, so theoretically there is an option to work with both programs. Practically, however, EPS files are not displayable on screen, so sizing and cropping of the EPS images are impossible. Also, EPS images will only print on Postscript devices, so if you don't have access to a Postscript printer, you are still out of luck for making the two programs work together.

The few problems notwithstanding, PageStream is one powerful and capable desktop publishing program. Soft-Logik has demonstrated a commitment to the program that has seen it through three major and much-needed upgrades in a relatively short period of time. With that track record, I expect that work will continue to squash the few remaining bugs and to improve an already powerful program. I would not be afraid to use this program on a regular basis anymore to do production desktop publishing. With its use of master pages and paragraph tagging functions, it has already become my desktop publisher of choice for large document production.

•AC•

PageStream 1.8
Soft-Logik Publishing Corp.
11131 S. Towne Square, Suite F
St. Louis, MI 63123
Inquiry # 204

upgrades • fixes • updates • new releases

bug bytes

by John Steiner

STEVE TIBBETT, AUTHOR OF VirusX, one of the most useful virus detection programs available for Amiga owners, announced via electronic media that a bogus version of VirusX recently appeared. The phony program is found under the name "VirusX 4.4". According to a posting on PeopleLink left by Mr. Tibbett, VirusX 4.4 appears to be a slightly modified VirusX 4.0 archive. The 4.4 archive contains a new "Virus" entry appended to the docs and has a longer description of this virus appended to the source file. Also, some punctuation in the source file was altered, and the 4.00 version number was patched to 4.40.

By the time you read this, there may be another genuine release of VirusX available that includes a few new features and the ability to detect some new viruses. It will not be numbered Version 4.4, so there shouldn't be any confusion. If you have Version 4.4 VirusX, don't give it to anyone else. Destroy it. VirusX is a public domain utility, and can be found on most BBS systems and information services that have Amiga software, or you can pick it up from a friend at your next Amiga Users Group meeting.

WORDPERFECT HAS A COUPLE OF bugs that can cause loss of data if you are not careful. As an avid user of WordPerfect, I have run across a couple of bugs that are related to the List Files requester. I have also learned to use the automatically timed backup which can be activated when you first set up and install the program.

The automatic backup has saved my work for me on several occasions. If you use the search option (number 9) of the List Files requester, it works just fine to highlight the names of the files that contain the target search. However, the program seems to stop searching at the first space in the

search text. In other words, the function appears to find only the first word in the search string requester. Once the files are highlighted and you try to search these files for additional words to narrow the search criteria, the program crashes.

Another less dangerous bug—and a workaround for it—was posted on PeopleLink. If you want to use the List Files requester to print multiple files and highlight several files to print, and the files are in a subdirectory instead of the root directory, the program cannot find the files. It notifies you of this fact for each file you have highlighted.

To work around this problem, simply copy (choice 8 on the List Files requester) the marked files to RAM:, then batch print from there. According to the person who left the posting (thanks Destiny), this problem and its solution were acknowledged by WordPerfect's technical support.

THE SAGA OF MAXIPLAN III AND Intuitive Technologies continues. In this month's mailbag, I received more bug reports from readers regarding MaxiPlan III. William Statler of Richland, WA sent a copy of a very detailed 6-page letter, complete with screen dumps, delineating several major problems in MaxiPlan III, Version 3.5. Bugs reported include problems with the graphics routines and a major bug that can actually result in the spreadsheet generating incorrect data—an unforgivable fault in a spreadsheet program. The problem with the generation of incorrect data occurs in minimum recalculation mode, when some cells fail to update. When cell names are used in formulas and the formulas link cells in a certain way, not all cells are updated properly. If you use minimum recalculation mode, be sure to test any data thoroughly.

Another MaxiPlan III bug reported involves the saving of a spreadsheet when more than one sheet is active. Occasionally during the save, the title bar of the spreadsheet being saved will contain the file name of the other active spreadsheet. According to the letter, "The spreadsheet is actually saved to the correct file, and immediately after the save, the title bar is corrected. Nevertheless, this bug can cause substantial terror as one sees one's file apparently being overwritten." Mr. Statler goes on to say that he has paid nearly \$250.00 for the original program and updates, and every version he has used has had numerous bugs. Many of the bugs have continued to exist in several versions, even though he and others have reported them.

Other correspondence has referenced additional problems found when using the latest MaxiPlan III upgrade. These people also complained of the non-responsiveness of the people at Intuitive Technologies. In early April I again placed a call to their customer support number, and left a message on the answering machine. I have now left two messages and have yet to receive a reply, even though Intuitive Technologies' recorded message assures users that their calls will be returned.

I RECEIVED A LETTER VIA CompuServe EMail from Tom Gist regarding a flaw in all versions of M2Sprint. Quoting from his letter:

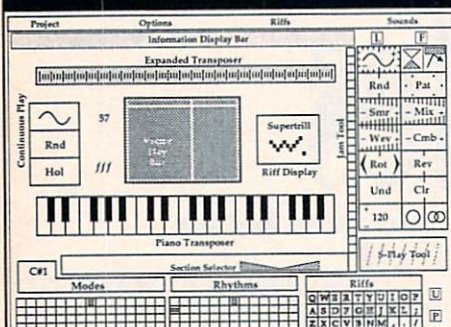
"There is a problem with the M2Sprint Modula II compiler's handling of real and longreal numbers. For instance, the code fragment

```
x := 0.05;  
WriteString(x := ');  
WriteReal(x,15);  
WriteLn; produces
```


THE DYNAMIC RIFF SEQUENCER

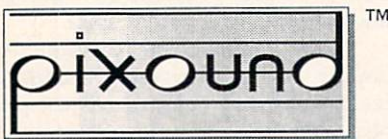


by Hologramphone Research



Turn your Amiga into a powerful new instrument with **Hyperchord™**, the dynamic riff sequencer. Create themes, from simple scale runs to complex "Riff Waves," using original Hg functions such as Smear, Rotate, Weave, Reverse, and Mix. Change pitch, speed, rhythm, harmonies and orchestration. For intense riffing, switch between 60 user-defined scale modes and 40 rhythms, or employ unique cyber-musical tools such as Holistic Play and Vector Play. Store for real-time playback or record performance. Disk includes three Hyperchord utilities: *Mode Maker*, *Rhythm Maker*, and *Holistic Window*.

Hologramphone's

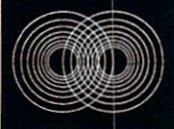


THE MUSICAL GRAPHICS PLAYER

Listen to a Lichtenstein!

Pixound™ is new kind of musical instrument as well as a powerful MIDI controller (uses Amiga sounds too). Load up any graphic image or use Pixound's screen generators. Invent a new instrument with every screen, then play it with the mouse. Create shimmering bursts of notes or slow, lyrical harmonies with the touch of a key. Save your work either as a musical sequence or a screen—or both. Great fun for the beginner; endless challenge for the virtuoso.

Circle 109 on Reader Service card.



Hologramphone Research
6225 S.W. 145th Street
Miami, Florida 33158

x := 0.5 instead of
x := 0.05

I've been trying since mid-Jan 90 to get Preferred Technologies to fix it and get me a corrected copy. After extreme pressure...Leon Frenkel sent me version 1.12. Unfortunately, his technical people knew in January that 1.12 did not fix the problem. Furthermore, he stated, 'You originally purchased M2Sprint from a company called M2S, Inc. We subsequently acquired the M2Sprint product from M2S, but we did not assume any obligations or liabilities [sic] of M2S. Also please note, almost all software published today has a disclaimer specifying that it is sold "as is" and is not guaranteed [sic] in any way. Even though we are not legally obligated we will always try to help a customer who is having a problem with any of our products.'

Mr. Gist goes on to say that further assistance was not forthcoming, and that he would "hate to see someone spend a sizable amount of money to get M2Sprint, thinking they could use it to perform real number calculations (as I did), and then discover that it doesn't work, and Preferred Technologies is not fixing it."

I placed a call to Preferred Technologies and talked to Mr. Frenkel. He commented that he sent the 1.12 upgrade to Mr. Gist and waived the normal upgrade fee. He said that he was not aware of the reason for the upgrade, but sent the upgrade as a customer courtesy. He said he was not aware of any problems with the program incorrectly handling real numbers, but if Mr. Gist would write with the details of the problem, he would look into the matter.

QUARTERBACK, FROM CENTRAL Coast Software, was one of the first hard disk backup utilities for the Amiga. The company has sent notices to registered users of Quarterback that Version 4.0 is now available. Several months ago, you may recall a bogus version of Quarterback was posted to several electronic bulletin boards. This program was a pirated, modified copy of Version 2, and was not a valid release from Central Coast Software.

So end users will not confuse the legitimate update with this bogus version, CCS has skipped over Version 3.0, and gone directly to Version 4.0. The company announced that there is no support for any version of Quarterback between 2.3 and 4.0. The latest official version contains several new features including: available hard disk volumes displayed for easy selection, visual progress bar (fuel gauge) added to volume status boxes, Print Catalog added to menu, volume/device name of

hard disk appended to archive/restoration report filenames, Restore Full Subdirectory Structure option improved, volume number added to archive/restoration reports, volume number added to QB backup volume disk ID, included/excluded status of directory shown on catalog, wrong volume error messages expanded during restoration, backup volume "test" mode added, file count display increased to 5 digits (99,999), and volume count increased to 3 digits (999).

Several minor bugs were also repaired in this release. It no longer leaves read locks on subdirectories, no longer causes the Amiga to crash after Quarterback exits, and no longer crashes upon loading the second disk when the alternate catalog is split across two disks. The upgrade is only \$10.00 including postage, and you must send them your original serial-numbered Quarterback diskette. **Contact:** Betty Chamberlain, Central Coast Software, 424 Vista Avenue, Golden, CO 80401, (303) 526-1030, Inquiry #200.

GOLD DISK IS EXPECTED TO HAVE released Professional Draw version 2.0 by the time you read this. Information posted to PeopleLink indicates that the upgrade price is set at \$60.00 and can be ordered immediately. Major improvements to Professional Draw include the ability to create text that follows a curved line, and much faster operation. **Contact:** Gold Disk, Inc., Box 789, Streetsville, Mississauga ON Canada L5M 2C2, (800) 387-8192, Inquiry #201.

MICROBOTICS ANNOUNCED AN upgrade to their HardFrame disk installation utility, RDPRep. The major advantage of the upgrade is an entirely "point-and-click" graphic interface that is much easier to use than previous versions. You may receive the upgraded version by either downloading it from the MicroBotics user support conference on BIX, or by sending \$7.00 to Microbotics and requesting the new HardFrame Installation disk. **Contact:** MicroBotics, Inc., 811 Alpha Drive, Ste. 335 Richardson, TX 75081, (214) 437-5330, Inquiry #202.

That's all for this month. If you have any workarounds or bugs to report, or if you know of any upgrades to commercial software, you may notify me by writing to:

John Steiner
c/o Amazing Computing
Box 869 Fall River, MA 02722 ...
or leave EMail to Publisher on PeopleLink
or 73075,1735 on CompuServe

•AC•

DIGIMATE III

Animation
Comes
To
DigiPaint

By Frank McMahon

DIGIMATE 3 IS A NEW ANIMATION UTILITY PROGRAM WHICH, AMONG OTHER things, lets you animate using DigiPaint 3. Before we get into the features, there are a few requirements that you must have in order to make full use of this program:

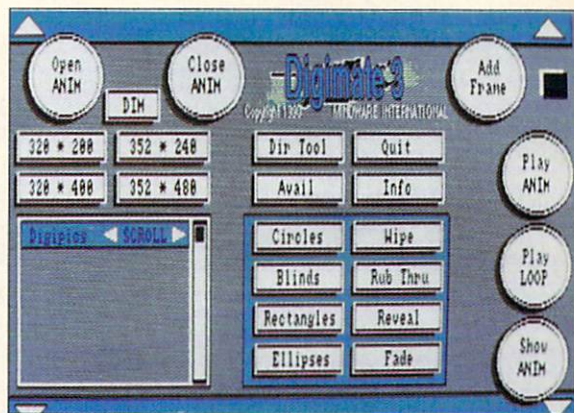
DIGIPAIN 3

While there are powerful animation processing utilities included, the main part of the program is creating ANIM files using DigiPaint3. While most utilities can work with DeluxePaint III animations, the center of the program is controlling DigiPaint 3 "frames".

AREXX

You may have heard of this program and not known what it was, but when Workbench 2.0 comes out, it will be as familiar as "IFF". ARExx is becoming a standard programming language, one that will be incorporated into the new version of Workbench. If a program such as DigiMate 3 or DigiPaint 3 supports ARExx, it means the program has a "port" or a "backdoor" which allows users and programmers to actually control the program! DigiPaint has the "port" in this case and DigiMate is the "controller", which is how DigiMate animates DigiPaint graphics.

What does all this mean to you? ARExx is a "transparent" program in this case, and while you do need a copy of the ARExx program in order to run DigiMate, you do not need to have any programming skills—you need only to look up how to install the ARExx program. It is very easily done, I might add.



All the DigiMate controls can be accessed from the main control screen.

Attention Readers!



AC WANTS YOU!

Since we at Amazing Computing™ cannot determine the dependability of our advertisers from their ads alone, we want your feedback. If you have had a problem with an advertiser in AC™, let us know! Send a complete description of your exchange(s) with the advertiser—along with the names of the individuals involved—and we'll do our best to get to the bottom of things.

Ad Complaints
PIM Publications, Inc.
Amazing Computing
P.O. Box 869
Fall River, MA 02722

2 DISK DRIVES

This program accesses your Workbench, ARExx disk, DigiPaint disk, DigiMate disk, AND whatever disk you save your animations on! I deleted several files on a backup of my Workbench disk and squeezed ARExx on it. This helps, but does not eliminate disk-swapping. If all concerned programs are on a hard drive or in RAM at once, the system runs like a champ, but with disks at least 2 drives are highly recommended. You might possibly get by with one drive—but loose your sanity in the process!

1 MEG CHIP RAM/1 MEG FAST RAM

We are working with HAM animations and pics running between two programs, and that eats up chip memory fast. If you want to do interlace HAM animations, you need the 1 meg Agnus chip and as much RAM as possible (at least 1 meg). If you lack either, a smaller version called "Tiny DigiMate" is included which runs from the Workbench screen and omits some of the program's features.

Now that we know what the program needs to run, let's get into what it actually does.

The program is a collection of several utilities that provide animation cell creation, HAM transitions, image and ANIM processing, and even a tool kit to work on disk files. You noticed I mentioned "cell

creation" and you probably thought of Disney, or perhaps the process of drawing cartoon frames on a peg board. This is the concept of animating with DigiMate and it is important to note that the program does not "move" graphics around the screen for you. It does provide you with stacks of blank frames on which to draw the motion, frame by frame. It's not as tedious as it sounds at first, and the program helps with different features such as automatic copying to next frame. So all you do is add the animating part and click on "Add Frame".

Once you have started the DigiMate program running your next step is to start up DigiPaint. Both programs will be running during the animating, and you will switch between them via "front to back" gadgets. It is the first time I have seen built-in multitasking in an animation program!

ON-LINE HELP

The main DigiMate screen contains various buttons which only require a mouse click to activate. One excellent feature is that if you position the mouse over any button and hit the "Help" key, context-sensitive on-line help appears (which can even be edited by the user). The help screens are part of the T.A.S.S. system DigiMate runs under. T.A.S.S. stands for "Thut Application Support System", an application development architecture framework which is incorporated into this (and other) Mindware products. T.A.S.S. creates an environment different from standard Workbench routines (such as vertical-scrolling windows and pop-up menus). I have found this framework much faster and easier to use than Workbench (pull-down menus, etc.) and that difference adds up to more productivity. Like ARExx, T.A.S.S. is an up-and-coming system you will be hearing more about.

CREATING A DIGI-PAINT ANIMATION

Once both programs are running, to switch between the two you need only hit the "screen back" button, which brings either program to the front. DigiMate can be positioned automatically at one-third height so even though it is in front you can still see your DigiPaint graphic.

After you create your first frame (which can be a stamped-down brush, a HAM graphic, etc.) click on "Open Anim". The previewing of your animation will be played directly off your device through the use of an included feature called "DiskAnim". This plays a frame at a time as soon as the frame loads. As you might expect, it's pretty slow from a floppy but somewhat faster off a hard drive. The best choice is saving to RAM disk.

Once your ANIM is opened, the first frame is automatically saved and you are returned to the DigiMate screen. Clicking on the "screen back" gadget brings the DigiPaint screen up front. Now you create the second frame. You may either pick up your brush and move it slightly (DigiPaint's coordinates should be on during this procedure), or draw in your next graphic or "cell". Traditional cell animation can be tough because there is no tracing or "auto-bluing" like in "Zeotrope" for example. This would provide an outline of the previous frame to go by. I recommend using DigiMate with a genlock and tracing cells that way, or better yet, digitize them and load them into the program. All DigiMate shows is the last frame created, which can be drawn over or altered.

Once you have created your second frame, switch to DigiMate and click on the "Add Frame" button. Repeat this as many times as needed and choose "Close Anim" when you are done with your animation. It is that easy. Preview your ANIM at any time by clicking on the "Play Anim" button. This allows you to check details such as speed and smoothness. There is also a "Play Loop" button for continuous play. A "ShowAnim" button with speed control is on the main screen. To use it, however, you must first obtain a copy of the Sparta Inc. public domain program and copy it to your "C" directory.

HAM TRANSITIONS

DigiMate also provides tools to create transitions from one HAM graphic to another. Both graphics must be of the same resolution, and both DigiMate and DigiPaint provide tools to convert graphics to the correct size. With the "DIM" gadget in DigiMate you can even enter in the exact screen width and height, which helps in

bringing in overscan graphics of various sizes. I have found overscan graphics work with no problem.

Transitions include circles, wipes, blinds, rub throughs, reveals, fades, rectangles, and ellipses. Some require two graphics, while others—such as fade—only require one. Additional transitions can be created using "Ed" on your Workbench to alter the ARexx macros. There are 6 additional ARexx macros to get you started. You can even alter your DigiMate screen to include your new transitions by loading the screen into Deluxe Paint III and changing the text on a gadget!

After you load your two graphics into DigiPaint, select the transition, and tell the program the correct resolution (there are buttons for regular-sized graphics, and "DIM" can be used for inputting overscan dimensions), you type in a filename. When you hit return the program begins to generate the transition. With HAM graphics it can take a while to render frames, but short 10 frame transitions take around 5-10 minutes. After the process is completed you have a standard ANIM of the transition that can be loaded in and played, or altered.

IMAGE AND ANIM PROCESSING

Until now, we have been saving ANIMs to disk or RAM and playing them from there. Through the "DigiPics" window we can load graphics or animations into RAM and process them. Many options are available, including:

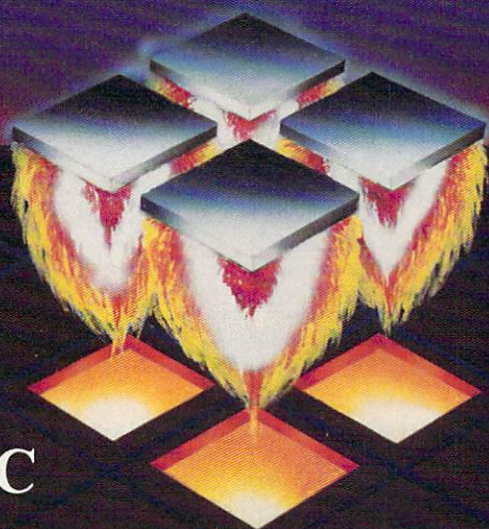
ANIM SPLITTER

Takes an existing animation (which, through "Delta compression", is only the "changes" from frame to frame) and splits it into separate IFF graphics. It even adds numbers to the root name automatically.

RESOLUTION/BIT PLANE CONVERSION

Changes images and animations from HAM to Halfbrite, HAM to LoRes, Convert to Black and White (Bit-Plane selectable), Toggle HiresBit and Videoescape to DeluxePaint.

Make Your Programs Take Off With Lattice C



Create the fastest programs in the shortest time with the high-powered Lattice® C Development System! Lattice's optimizing C compiler takes your code and generates lightning fast programs. Hundreds of library functions and more than a dozen utilities help you build your programs in record time.

Order the Lattice C Development System and you get:

- ◆ Optimizing C Compiler
- ◆ Source Level Debugger
- ◆ Macro Assembler
- ◆ Overlay Linker
- ◆ Lattice Screen Editor
- ◆ Compiler Companion Utilities
- ◆ Code Profiler
- ◆ Disassembler
- ◆ Extensive Documentation
- ◆ Unlimited Free Support

Lattice offers a complete line of programming tools including our C++ compiler, MS-DOS to AmigaDOS C Cross Compiler, Libraries, Utilities, and C programming seminars. Call or write for a free catalog.



Lattice, Inc.
2500 S. Highland Ave.
Lombard, IL 60148

Phone: 1-800-444-4309
708-916-1600
FAX: 708-916-1190
Telex: 532253

Lattice

Subsidiary of SAS Institute Inc.

Circle 124 on Reader Service card.

SPECIAL TRANSITIONS

More transitions applied to an existing animation include: Half-height/Mirror, FlipXAxis, and Half-Width/Mirror.

EDGE DETECTION

Allows you to reduce an image to its component boundaries between parts of the image.

ANIM CONSTRUCTING

As it implies, this process takes a series of IFF graphics or an animation you have split into separate frames, and combines them into an animation.

CHOP LAST FRAME

This is for handling DeluxePaint III animations, which adds copies of the first



Exciting New Game
of
Legal Affairs

Courtroom

- Act as Prosecutor or Defense Attorney
- Play against the Computer or Another Person
- Choose from Liberal or Conservative Judges
- Select Criminal Cases from the Court Docket
- Question Witnesses, Raise Objections
- Convince the Jury and Win the Case

from...

only **FairBrother & SoeparMann**
\$49.95 5054 S. 22nd Street
Arlington, Virginia 22206
(703) 820-1954

Shipped UPS Ground. COD or UPS 2nd Day Air Add \$3.00

Circle 179 on Reader Service card.

two frames on to the end of an Anim. These frames must be removed before you add any frames on using DigiMate.

Lastly, the T.A.S.S. system features an extensive program called "DirTool". It makes heavy-duty file work very, very easy, allowing direct access to AmigaDOS commands such as "path", "assign", and others. Since this is an added feature and not really an animation utility, I won't get too much into it except to say it is a fantastic program that allows you to bypass the CLI.

CONCLUSIONS

DigiMate III claims to provide animation to DigiPaint III and that's exactly what it does. Traditional "cell" animation but not automatic "moving" like DeluxePaint III. The edge this package has over DeluxePaint is the few thousand extra colors, which can be used to create some dazzling effects. The package is not without its few problems, though.

I would like to be able to turn off the huge "busy" icon when an animation is playing from disk. It doesn't seem to support Animbrushes (and doesn't claim to). The DigiMate screen is on the lower third of the screen, and if you get a system request, that also appears on the lower third. Of course,

you can't hit "cancel" or "retry" if you need to, because they are both off the bottom of the screen. And you can't raise the screen until the request is completed! The built-in HAM transitions are fine for home use, but not smooth enough for professional use.

Altering them through ARExx macros would cure this, but that should not be necessary. Image processing on HAM animations can take a while, especially on longer ones, and there is no abort key.

Also the "DiskAnim" feature promises to play animations from hard drives and "break the RAM barrier!" Well it does, but don't get too excited.

Amiga* Digest Video Series

Tape 1 - Mastering Workbench* and CLI*

Step-by-step guide on how to get the most from the Workbench and CLI environments. FREE PD software disk and command summary.

Tape 2 - DeskTopPublishing with PageStream**

Complete instruction that takes you from start to finish on two projects. FREE fonts/clipart disk and keyboard command summary.

\$30 each or both for \$50

Includes UPS shipping VA res. add 4.5% tax

Call for Free Discount Coupon and Product Guide

Grass Roots Video Productions

P.O. Box 10889

Burke, VA 22015 (703) 569-2652

MasterCard, VISA, Check, MO, COD

*Trademark of Commodore Business Machines
**Trademark of Software Publishing Corporation

Circle 108 on Reader Service card.

It plays at a constant but slow pace, maybe 1 or 2 frames per second, and could only be used for previewing and never for dumping a real-time 10 meg animation to video tape. Double-buffering and preloading several frames could eventually make real-time hard drive animations a reality, but "DiskAnim" is only a preview tool. It does play much faster from RAM, as you might expect.

On the plus side, the whole system of creating frames is easy to use and the T.A.S.S. control system is far superior to most animation control panels. You can relax and have fun

animating and not get bogged down in menu commands. The system is very natural and not too busy or clunky. HAM transitions have no fringing, and combining different palette images is okay, too. If you want to get into text editing, create hundreds of your own custom transition effects and add them to your main screen as buttons. While converting images to different resolutions and bit planes is nothing new, using those tools on animations is—and DigiMate does this effortlessly.

Customer support is excellent. I called just to discuss the program and possible additions, and found Mindware to be extremely helpful and very willing to listen. Their automatic update program gets high marks as well.

If you have DigiPaint 3 and ARExx and want to do HAM animations, this is an excellent low-cost choice.

If you don't have ARExx, the cost of this package doubles because you must purchase ARExx separately, at least until Workbench 2.0 comes out. The DigiMate box includes a discount offer for the latest version of ARExx. A small plug must go to William S. Hawes, whose ARExx language is allowing programs like these to talk and be controlled by one another. As the first wave of these type programs begin, DigiMate III adds a powerful feature to an already powerful program—Animation!

•AC•

Memory Management Amiga Service Specialists

Over three years experience!
Commodore authorized full service center. Low flat rate plus parts.

Proudly affiliated with . . .

The Memory Location
396 Washington Street
Wellesley, MA 02181
(617) 237-6846

Circle 186 on Reader Service card.

The Am512

512K Memory/Real Time Expansion Board

For the Amiga 500 Computer

INTRODUCTION

THE AMIGA REQUIRES LARGE AMOUNTS OF MEMORY BECAUSE OF ITS superior graphics and multitasking abilities, and the basic Amiga 500 comes with 512K bytes of dynamic memory. This memory is shared between the custom graphics chips and the MC68000 microprocessor. It is also called the "chip memory", since the custom chips have access to it. This project will allow you to double your memory to 1 megabyte and add a real time clock to your Amiga 500 computer.

The serious user soon learns that the basic 512K isn't enough. Some of the advanced games on the Amiga require at least 1 megabyte of memory and you will be seeing more of them in the future. The new "Super Fat Agnus" chip will also open the door for more development of software that requires at least 1 megabyte of memory.

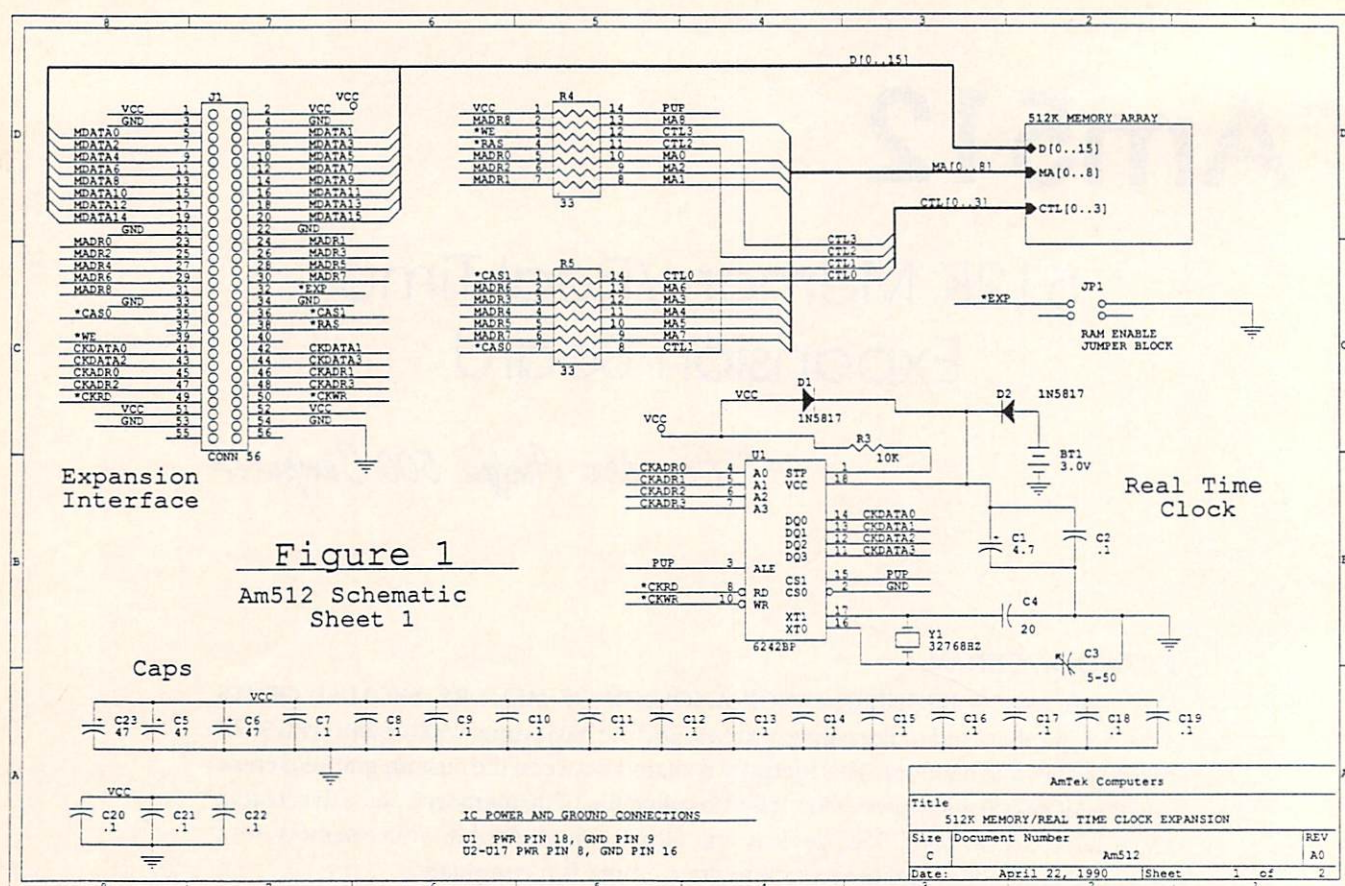
Upgrading the Amiga 500 to a 1 megabyte machine is quite easy, since Commodore provides a built-in interface. The 512K memory expansion has been called the "trap door memory", since the expansion memory installs inside the small door located at the bottom of the A500 computer. The memory expansion also supports a real time clock—this allows the A500 to automatically load the date and time when it boots up. This expansion memory is located at address C00000h to C7FFFFh in the memory map. This memory is called "fast RAM" since the custom chips don't have access to it, and therefore can't cause contention when the MC68000 microprocessor uses it.

The new "Super Fat Agnus" custom chip allows this memory to be used by the custom graphics and sound chips. This doubles the amount of memory used for screen graphics and sound applications. It also turns this "fast RAM" into "chip RAM", since the custom chips have direct access to it. Those with the "Super Fat Agnus" custom chip that want "fast RAM", need expansion memory located in the 8 meg space starting at address 200000h to 9FFFFFFh.

The memory board described in this article, the Am512, is functionally equivalent to the A501 memory/real time board built by Commodore with one exception: the

by James Bentley





Am512 memory/real time expansion board powers the real time clock with a lithium battery instead of a NICAD battery. This is important for two reasons:

First, the lithium battery will last for 10 years, while the life cycle of the NICAD battery varies between 3-5 years. You therefore double your battery life with lithium batteries. The battery on the Am512 is socketed for easy replacement and can be purchased for about \$5.00 at any fine photography shop.

Second, there is no power drain on the +12 volt power supply. The lithium battery doesn't need to be recharged. Commodore uses the +12 volt supply to recharge the NICAD battery pack, while the main power is on the A500. The NICAD's don't require much current for recharging, but the power supply on the A500 is rather limited and when you start adding extra peripherals like floppy drives and expansion memory, all the power counts.

The Commodore A501 memory/real time expansion also includes an EMI shield that totally encases the circuit board. This makes battery replacement very difficult and probably shouldn't be attempted by the average hobbyist. I recently asked an employee of an authorized Commodore dealer/service store how to replace the battery on the A501. He stated that "the battery never needs replacement"—I guess he thinks a NICAD battery lasts forever! Regardless, the EMI shield is soldered to the circuit board and battery replacement should be left to an authorized service center which is guaranteed to cost you more than the cost of the NICAD battery alone.

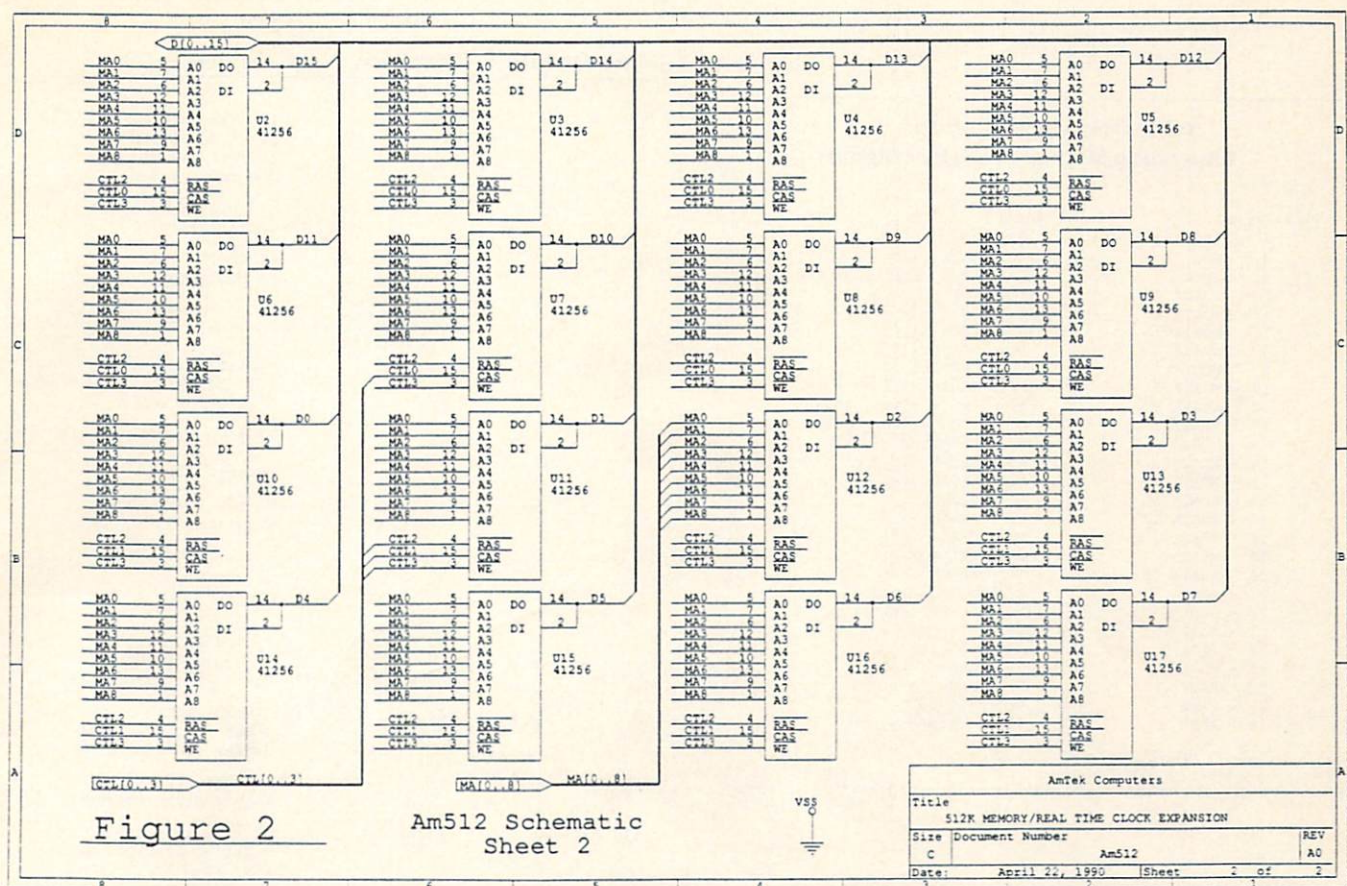
The Am512 memory/real time expansion described in this article doesn't include an EMI shield, and it isn't FCC approved. Initial tests show that it doesn't cause any EMI problems, but

construction and use of the Am512 board is the responsibility of the end user. Now let's get to the details.

CIRCUIT DESCRIPTION

The Am512 interfaces with the A500 computer through the memory expansion port located at the bottom of the A500 computer. This expansion port is a 56 pin connector built-in by Commodore. Refer to Table One for the signal names. In general, the signals required for the 512K dynamic memory array are numbered 5-20, MDATA(15:0), I/O data lines, numbers 23-31, MADR(0:8) DRAM address lines and the control lines numbered 35-39, *WE, *CAS0, *CAS1, and *RAS. Those familiar with dynamic memory know that it requires refresh cycles about every 15 microseconds, so the data is retained inside the chip. The address multiplexing and refresh is controlled by the "Fat Agnus" custom chip inside the Amiga. This is the main reason why building this memory expansion is so easy. The 512K dynamic RAM array consists of (16) 256K x 1, 150 nanosecond memory chips. These memory chips come in a 16-pin DIP package and have recently dropped in price to the \$2.00 level.

The signals required for the real time clock are numbered 41-44, CKDATA(0-3), I/O data lines, numbers 45-48, CKADR(0-3), address lines and the control lines, numbers 49 and 50, CKRD, read and CKWR, write signals. The custom chip "Fat Agnus" controls the read/write signals. The real time clock chip is the MSM6242 built by OKI Semiconductor. This is the same one used on the A501 board. This chip features a 4-bit data bus and 4-bit address bus, an auto leap year adjustment, 32.678k hertz crystal-controlled time base and CMOS construction that requires +5 volt only. The real time



chip provides seconds, minutes, hours, day of week, day, month and year to the Amiga operating system for time/date keeping. The MSM6242 allows standby operation down to 2.0 volts and draws less than 15 microamperes in standby mode. This allows the lithium battery to power this chip for 10 years or more in standby mode.

Figures One and Two are the schematics of the Am512 memory/real time expansion board. The 512K memory array consists of (16) dynamic memory chips, U2—U17, along with a bypass capacitor for each memory chip. Resistor packs R4 and R5 provide under voltage damping for the address lines to the memory array. Address multiplexing and all control for the dynamic RAMs is controlled by "Fat Agnus" on the A500 mother board and therefore will not be discussed here. U1 is the real time clock chip, notice Y1, 32.768k hertz crystal, along with C3 and C4 provide the tank circuit for the crystal time base. C3 is adjustable and can be used to fine tune the accuracy of the time base. Diodes D1 and D2 are Schottky barrier diodes that feature very low forward voltage drops and very high reverse leakage currents. The diode circuits allow the real time clock chip to be powered from the Amiga when the main power is on and then automatically switch to the lithium battery backup when the main power is turned off. Battery B1 is the lithium battery built by Sanyo, part number CR17335SE. This is a 2/3A size, rated at 3.0 volts, 1.3 ampere/hour and can be substituted with an equivalent battery. Connector J1 is the 56-pin dual row connector for interfacing with the A500 computer. Jumper JP1 allows you to disable the 512K memory array when using the real time clock function of the board only. The rest of the components are capacitors used for filtering the power supply.

CONSTRUCTION

It is recommended that a printed circuit board be used for construction of this project, but it also could be wire-wrapped (a silk screen for a printed circuit board, supplied by the manufacturer, is listed in the parts list). If it is wire-wrapped, make sure you use a board with a solid power and ground plane, and use short wires to connect the power and ground to the memory chips to reduce noise and ground bounce switching. (The printed circuit board supplied with the kit, has all the parts placements marked with the silk screen.)

All components can be soldered directly to the printed circuit board, or the memory array could be socketed for easy replacement of a defective memory chip. Notice pin 1 by the square pad for all IC's. Connector J1 must be mounted so pins can be inserted from the board edge side. Diodes D1 and D2 have polarity and are marked by the dark band at one end of the diode. Capacitor C3 should be mounted with the flat side toward the closest edge of the board. The battery socket has no polarity, but when inserting the battery please notice the large plus sign on the board and match it with the plus sign on the battery. All the electrolytic capacitors have polarity and the printed circuit board is marked. The rest of the components and bypass capacitors have no polarity and can be mounted either way.

After construction, please double-check all components on the Am512 board before proceeding to the installation of the Am512 board.

**Table One Amiga Built-In
Expansion Memory Interface Signals**

Pin 1 - +5 volt	Pin 29 - MADR 6
2 - +5 volt	30 - MADR 7
3 - Gnd	31 - MADR 8
4 - Gnd	32 - *Exp (enable memory)
5 - MDATA 0	33 - Gnd
6 - MDATA 1	34 - Gnd
7 - MDATA 2	35 - *CAS0
8 - MDATA 3	36 - *CAS1
9 - MDATA 4	37 - *RAS0
10 - MDATA 5	38 - *RAS1
11 - MDATA 6	39 - *WE (write enable)
12 - MDATA 7	40 - *CCK clock (3.58Mhz)
13 - MDATA 8	41 - CKDATA 0
14 - MDATA 9	42 - CKDATA 1
15 - MDATA 10	43 - CKDATA 2
16 - MDATA 11	44 - CKDATA 3
17 - MDATA 12	45 - CKADR 0
18 - MDATA 13	46 - CKADR 1
19 - MDATA 14	47 - CKADR 2
20 - MDATA 15	48 - CKADR 3
21 - Gnd	49 - *CKRD (read)
22 - Gnd	50 - *CKWR (write)
23 - MADR 0	51 - +5 volt
24 - MADR 1	52 - +5 volt
25 - MADR 2	53 - Gnd
26 - MADR 3	54 - Gnd
27 - MADR 4	55 - +12 volt
28 - MADR 5	56 - not connected

Note: * Active low signal

INSTALLATION

The Am512 memory/real time clock board includes a jumper strap to enable/disable the entire 512K memory array. This allows the Am512 board to be used as a real time clock only. Locate the jumper strap JP1 marked on the printed circuit board (upper left corner). With the strap in the top position, the 512K memory array is enabled. With the strap in the bottom position the 512K memory array is disabled.

The Am512 installs in the bottom of your A500 computer. Make sure the power is off before installing the Am512 board. Carefully remove the plastic door located at the bottom of your computer, using a flat screw driver or coin to pry the door open.

Install the Am512 board with the components facing the computer. Carefully align the connector at the end of the Am512 board with the pins inside the computer. Slide the Am512 board into place but don't force it, because you may damage the computer or the Am512 board. The Am512 should seat firmly into place. Check the alignment and replace the plastic door before supplying power to your computer.

The A500 computer will automatically recognize the additional memory and begin to use it at boot up time. To set the real time clock for the first time, use the DATE command to set the date and time, then use the SETCLOCK RESET and SETCLOCK SAVE commands of AmigaDOS. Your STARTUP-SEQUENCE file in the s/ directory should include the SETCLOCK LOAD command, so the Am512 real time clock will be loaded each time you boot the A500

Am512 Parts List

Item	Quantity	Reference	Part
1	1	BT1	Lithium Battery Sanyo CR17335SE Duacell DL2/3A 3.0 volt, 1.3 A/H
2	1	C1	4.7 uf, 35 volt Radial Cap
3	17	C2,C7,C9,C10 C11,C12,C13 C14,C15,C16 C17,C18,C19 C20,C21,C22	.1 uf, 12 volt Ceramic Caps
4	1	C3	Variable Capacitor ERIE TZ03Z500 5-50 pf. NPO
5	1	C4	20 pf., 50 volt Ceramic Cap
6	3	C5,C6,C23	47 uf, 35v, Radial Capacitors
7	2	D1,D2	1N5817 Schottky Diodes
8	1	J1	56 Pin dual row connector Method 9008-228-302 Cannon UBS4A056A1DL Elco 8251-056-000-826
9	1	JP1	Jumper block Samtec TSW-102-14-LD Jumper SNT-100-BKG
10	1	R3	10k ohm, 1/4 watt Carbon Comp resistor
11	2	R4,R5	33 ohm, 14 pin dip parallel resistor paks CTS 760-3-R33
12	1	U1	MSM6242BRS/GS real time clock chip, OKI Semiconductor, Inc.
13	16	U2,U3,U4,U5 U6,U7,U8,U9 U10,U11,U12 U13,U14,U15 U16,U17	256k x 1, 16 pin dynamic memory chips 150 nSec, 41256 (generic)
14	1	Y1	32.768k hertz clock crystal
15	11	-	2/3A size battery holder Keystone 131
Miscellaneous:		Printed circuit board, 16 pin dip IC sockets.	

computer. See your DOS manual for additional information on how to use the DATE and SETCLOCK commands.

Good luck, and have fun with the Am512 512K memory/real time clock board.

Note: The following items are available from AmTek Computers: A kit of all parts, including double-sided, solder-masked, silk-screened PC board and memory for \$69.95, same without memory for \$39.95. PC board alone for \$15.00. Lithium battery and holder for \$6.00. MSM6242 chip, 5-50 pF variable capacitor and 32.768 kHz crystal for \$8.75. 56-pin dual row connector for \$6.50. Add \$3.00 shipping and handling. Utah residents must include sales tax. AmTek Computers, P.O. Box 575713, 5136 Clover Meadow Drive, Salt Lake City, UT 84123. Phone: (801) 261-1115. Inquiry #273.

•AC•

CHERRYWARE

by Julie Petersen

Freeware, trojanware, wolfware...

Extend your vocabulary with computerisms that define products, and describe characteristics associated with their use.



WITH NEW TECHNOLOGIES COME NEW VOCABULARIES. IN FACT, COMPUTER words are evolving so quickly, that publishers are creating separate volumes to describe them, rather than trying to add them into regular dictionaries. One example is *Webster's New World Dictionary of Computer Terms*. In paperback form, this reference is over 400 pages long, yet even that doesn't seem to be enough space to include many common phrases or nuances of meaning.

If terms are too vague, misunderstandings can result. An example of this is the catch-all 'vaporware'. It fails to make a distinction between companies struggling to put the finishing touches on a product and those who are using deceitful practices to extract money from unwary buyers.

So, for your edification and enjoyment, here are some phrases which may not be listed in Webster, or the pages of IEEE journals, but which help describe some of the characteristics and legalities associated with computer-related products.

COMMERCIAL PRODUCTS

Commercialware is appropriately flagged with a price tag. There is a common myth that commercial software is cheap to produce. I've watched messages fly back and forth on bulletin boards, arguing that software companies are making a killing, based on the assumption that 'diskettes and a bit of photocopying only cost a couple of bucks'. In reality, publishers only receive about 40% of the list price, and this has to pay for small details like rent, wages, advertising, packaging, shipping, development, research, customer service, long distance calls, storage, upgrades, and capital for future projects.

NONCOMMERCIAL PRODUCTS

*'Public domain' is sometimes incorrectly used as a catch-all term for products which are distributed noncommercially. But since PD has a more specific legal connotation involving the author's rights, I'd like to suggest **everyware** as a blanket term for noncommercial software distributed hand-to-hand, through bulletin boards, user groups, or as 'cost of media plus shipping' products. Then public domain software fits more correctly as a subcategory of everyware:*

• **Public Domain.** The author relinquishes all rights and control over the software and makes it available to anyone to modify, extend, sell, or give away. Since the recent revision of US copyright laws and compliance with the Berne Convention, software remains the property of the author unless explicitly stated otherwise. Personally, I feel that even if a program is donated to the public domain, professional ethics, and respect for individuals suggests that they should be credited for their ideas and work. The situation gets ticklish; however, when someone takes public domain code, modifies it and then claims copyright over the new version. In court this becomes a complicated issue, usually influenced by the degree to which the code has been changed or extended.

• **Shareware.** Shareware is really commercialware that ends up being everyware. That is, the users of shareware often don't pay. Shareware is based on the 'try before you buy' philosophy. It also gives small developers the opportunity to get into the software business, since they bypass some of the advertising and packaging costs. Shareware is a superb concept intended to benefit everyone. However, it has only partially succeeded due to inertia, or reluctance to pay. A few shareware authors have received compensation for their work, but an informal survey indicates that most authors can count on two hands the number of people who voluntarily contribute. Recently, European and North American user groups have begun to coordinate collection of shareware donations, and forward them to deserving authors. This may help to keep the shareware concept alive.

• **Freeware.** Freeware differs from public domain software in that the author or publisher retains rights to the software, and distributes it on the understanding that it will not be altered or sold for profit.

• **Candyware.** A product which has some key features disabled. Candyware is distributed to give a 'taste' of the complete product, which is usually commercialware. It's based on the belief that the users are motivated to inquire further or purchase the product if they've become familiar with its features and use. Incentive software.

GENERAL TERMS

This section encompasses both commercialware and everyware, describing products according to their quality or utility:

•**Barleyware.** This is a product which works, just barely. During the depression years, barley water was cooked up as a substitute for hot coffee. It was thin, dull, and lackluster. Barleyware, like barley water, is generally bug-free, and gets the job done in the most minimal way, without any bells or whistles, but quickly fades away if a better product comes along.

•**Dismalware.** This software should never have left the computer that spawned it. It usually has a complicated or brain-damaged interface, poor to nonexistent documentation, and bugs so numerous, it needs fumigating. Dismalware is sometimes excusably distributed as PD by novice programmers hoping someone will take the idea and improve the implementation. This theory of evolution sometimes results in useful software. If it is distributed commercially; however, people tend to lean towards a creationist viewpoint, expecting the product to be in finished form at time of release. The best response to commercial dismalware is to boycott the product.

•**Funware.** Computer games and other software written and distributed for entertainment. Jollyware is a subset of funware including screen hacks, surprise endings, etc.

VAPORWARE

Vaporware refers to products which are discussed, demonstrated, or promoted publicly, but which are not generally available:

•**Beeware.** The worst category of vaporware. It is promoted as a commercial product simply to see if there is enough interest to justify development. The idea is still on the drawing board and money collected from orders may or may not be returned to owners if the project is never initiated. When purchasers try to contact the company for information, they are often put off with excuses, and refunds can be slow or absent. If you order beeware, you usually get stung.

•**Ventureware.** A nearly finished product being developed by an individual or company with insufficient funds to finish the job. Collection of money from orders sometimes succeeds in bankrolling the final stages and packaging of a product, thus bringing it to market. Ventureware is illegal unless purchasers are explicitly

informed that the product is not complete. Although illegal ventureware shouldn't be tolerated, it's also true that a lot of companies got started this way.

•**Nearlyware.** A product which is 'essentially' complete. It works in most circumstances and would be useful to many people, but is not bullet-proof enough to release to a general market. Unfortunately, the agony and time involved in making nearlyware bullet-proof pushes up the development costs and compromises the 'business window' during which a product can be competitively released. Assuming no release date announcement, we should probably be tolerant of nearlyware, the frantic developers are usually guzzling caffeine and working 120-hour weeks to complete the project.

•**Buggyware.** This is almost the opposite of barleyware. Whereas barleyware is an average concept reasonably well executed, buggyware is a good product which just isn't finished. Buggyware has so many bugs, it is virtually unusable. Companies releasing buggyware have usually had their judgement clouded by contract deadlines, diminishing funds, or competition. Buggyware sometimes results from misplaced faith in in-house testers. It's clear that people who are technically-oriented or intimately acquainted with the product do not ferret out the same quantities and types of bugs as the general public. In the long run, buggyware engenders more anger from users than other 'vaporware'. When people invest time, money, and trust in their purchase, they should receive a working product.

•**Neverware.** A good idea that died. Money, time, unforeseen problems, boredom with a project, frustration, competition, and even theft are all reasons why neverware never makes it there. It's unfortunate when someone steals an idea and implements it first, particularly if the implementation is bad. It's difficult too, if the competition gets to the finish line sooner. But a bigger share of neverware results from authors not truly appreciating the complexity of creating a complete product, or the amount of time involved.

NOWARE

This is software that has no business being distributed in any form. Includes pirate software (also called 'warez'), viruses, or any programs

which promote destruction or negative discrimination:

•**Wolfware.** Predatory software. Software that directly damages or devours data, especially on hard drives.

•**Wormware.** Invasive software. Insinuates itself into programs and/or drives. Damages memory and data indirectly by munching away at the available space.

•**Trojanware.** Trojan horse software. Programs that masquerade as others, or which lie in wait, ready to capture passwords, or other information which could be used to gain unauthorized access to a system or account.

•**Bunnyware.** Innocuous programs which display a cute message or picture and then sometimes go away, without compromising program execution or data. Although bunnyware seems harmless, it's in the noware category because the person who encounters it doesn't know if it's bunnyware or wolfware, and thus is inconvenienced by having to treat it as wolfware.

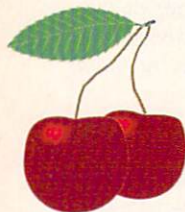
So what do we call exemplary products? Well, how about CHERRYWARE as a term reserved for computer wares that surprise and benefit us in many ways. Cherryware is bug-free, polished, intuitive, and loaded with features.

Although this article is light-hearted, there's always a serious side. There are at least three Amiga developers working on video, animation, and vector drawing products that are, well, nearly there. And, if we're a little bit patient, maybe we can look forward to some new cherryware.

•AC•

An active developer, consultant and writer, Julie Petersen has been an Amiga advocate right from the beginning. In September 1985, she formed PaNorAmA, the Pacific Northwest Amiga Association, one of the first Amiga groups, and served as its President for 2 years. Ms. Petersen has a long history of national and international recognition for her work in computer graphics and fine arts. In one of her more recent accomplishments, an interactive videodisc for which she created the graphics received a Canadian National Award of Excellence in 1989.

Send questions or comments to Julie Petersen, c/o Amazing Computing, P.O. Box 869, Fall River, MA 02722-0869, or she may be contacted on Usenet at portal!cup.portal.com!LadyHawke.



Convergence

Part Five of the Fractal Series

Now, don't
let the title
'Convergence'
scare you, it
is not a new
religion or
political action
committee.

UNTIL NOW WE HAVE BEEN DEVELOPING basic skills which could be applied to many different graphic programs. It is now time to start with the first of a series of mathematical concepts that will give us an understanding of the Mandelbrot set itself. Now, don't let the title "Convergence" scare you, it is not a new religion or political action committee. However, before defining it, let's talk a bit about equations.

SOLUTIONS OF EQUATIONS

So far whenever I have presented an equation, I have asked only that you perform a single calculation based on some given values of its variables. Let's do that one more time for the following equation:

$$y = x * x$$

I might ask you to solve the equation for some value of x , let's say $x = 3$. In that case, $y = 9$ is the answer. We say that $y = 9$ is the solution of the equation when $x = 3$ because that value makes the equation a true statement. Whenever x equals 3, y must equal 9, otherwise the equation would represent a false statement. But who says that x must equal 3? Nobody, really. x could equal something else, like 5 for instance, in which case the solution of the equation would be $y = 25$. We say that $y = 9$, and $y = 25$ are particular solutions of the equation when $x = 3$ and $x = 5$. Each one is a solution for a particular value of x . It turns out that x could equal any number whatso-

ever, and for every value of x , there would be one particular solution or value of y for which the equation was true. So the equation really has an endless number of solutions.

A SOLUTION SET

We call all the possible solutions of an equation collectively a solution set. It is the set of all possible values of each variable which form solutions of the equation. In our example the solution set would be a very long list of numbers:

$y = 1$ when $x = 1$
 $y = 4$ when $x = 2$
 $y = 9$ when $x = 3$
 $y = 16$ when $x = 4$
 $y = 25$ when $x = 5$
 $y = 36$ when $x = 6$
.....

The list is infinite!

Similarly, the Mandelbrot set is a whole bunch of solutions, but for a different equation, of course.

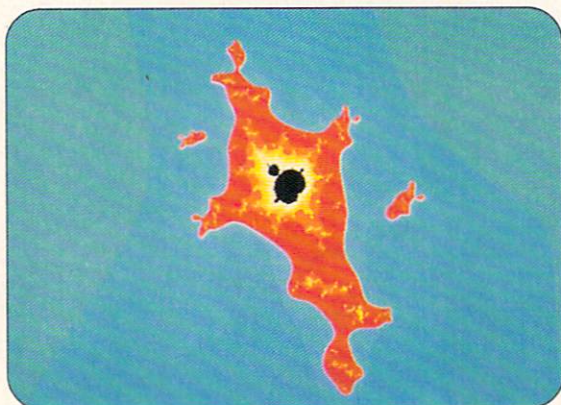
The above equation involves only two variables, x and y . I gave you a value for one and asked you to solve for the other. Other equations could involve more variables. You could for instance be given values for two variables, x and y , and asked to solve for a third, z . You could also be asked to find the solution set for a particular value of z , all values of x and y which represent a true statement for a particular value of z . In last month's equation:

$$x^2 + y^2 < 25$$

→

by Paul Castonguay

Figure One:
Generated
using
'Calculate'
subroutine
of Listing #3



all points falling within the circle represented a particular solution set of the equation:

$$x^2 + y^2 < z^2$$

where $z = 5$. That was the solution set of x and y for a circle of radius 5. Larger circles have larger solution sets.

CALCULATION MACHINE, INPUT AND OUTPUT

I want you now to turn your imagination up to full volume. When you see an equation I want you to think of it as a little miniature calculating machine living inside your computer program. If you give the machine some numbers (which we will call INPUT), it responds by performing calculations and returning an answer (which we will call OUTPUT).

SELF-FEEDING EQUATIONS

Let's beat this same example to death:

$$y = x * x$$

I will give you a value for x and ask you to calculate the corresponding value of y . Ready? Let $x = 3$, calculate y . The answer is $y = 9$, right? The INPUT was 3, the OUTPUT is 9. Ok,

now let $x = 9$, and calculate y again. The answer is $y = 81$. What did we do here? We took the OUTPUT from the first calculation and used it as INPUT to the second. OK, now let $x = 81$. Go get your hand calculator and confirm that the answer is $y = 6561$. Are you getting the idea? I am repetitively feeding numbers into the calculation machine, and each time I do, I use the OUTPUT from the previous calculation. The OUTPUT of one calculation is being fed back to the calculating machine as INPUT to the next. The calculating machine is sort of feeding itself. We've created a monster!

A SMALL MONSTER

Let's write a small test program:

```
x = 2
count = 0
Loop:
  y = x * x
  count = count + 1
  print "The answer is"; y;
  print "after"; count; " loops"
  x = y
GOTO Loop
```

Run the program. My Amiga reports that the answer is 1.844674E+19 after 6 loops. The screen flashes orange and a message appears at the top, reporting that an 'overflow' has occurred. I told you it was a monster! To regain control I had to first select STOP from the pull down menu and then click on the OK box in the overflow message. Don't worry, I did not damage my Amiga, and neither will you. But what is really going on here?

This program does exactly what we were doing above, feeding the calculation machine with the OUTPUT from the previous calculation. Only it does it much faster. After performing each calculation, it makes the INPUT variable equal to the OUTPUT variable. The important line is:

$$x = y$$

That is equivalent to assigning the OUTPUT of the last calculation to the INPUT of the next.

$$\text{next input} = \text{last output}$$

Then the program loops around to perform another calculation.

It turns out that we did not even have to use the variable 'y' at all. We could have used the line:

$$x = x * x$$

That way both the calculation and the assignment of the OUTPUT to the INPUT get done in the same line. In fact, there is even a special notation used for writing what we are doing here. I will tell you about it, because it will help you read those Scientific American articles called 'Computer Recreations', which you may be trying to understand. The notation uses a little arrow:

$$\begin{aligned} x(n) &\leftarrow x(n-1) * x(n-1) \\ \text{or} \\ x(n) &\leftarrow (x(n-1))^2 \end{aligned}$$

Figure Two:
Fractal using
'Calculate'
subroutine
of Listing #4



The term $x(n)$ represents the n th calculation out of a long series of unspecified length. The term $x(n-1)$ represents the previous one. Thus, if $n = 5$, we are talking about the fifth calculation, and it is using the result from the fourth as INPUT. We say that x gets the square of x from the previous calculation.

Incidentally, the magazine column 'Computer Recreations' appears in Scientific American every month, and it often covers ideas about fractals. Note however, that it is written at a higher technical level than these articles.

WHAT ABOUT THE ERROR?

But the Amiga did not like all this! It complained:

overflow error

Our calculating machine blew up! The numbers got bigger and bigger until the Amiga could not take it any more. What good is that? You'll see.

Now, activate the LIST window and change the first line of the program to:

$x = 0.9$

Run the program again. Hmmm.... It doesn't blow up this time. But it doesn't stop either. It simply goes and goes until the OUTPUT of each calculation equals zero. Over and over again, zero is fed back to the calculating machine, and over and over again, the answer is zero. How boring! Stop the program. (pull down menu or [CTRL-C]) You have just witnessed convergence.

CONVERGENCE

Convergence occurs when a series of calculations, repeated a large number of times, produces a single answer. In the above example, the answer became zero after only a few loops. After that, it remained zero, no matter how many times the loop was allowed to execute. The equation converged to zero!

The opposite of convergence is blowing up, which is what happens when the initial value of x is 1.5. You can call blowing up - convergence to infinity.

A FACT OF LIFE

But wait a minute. How is it possible that the same equation can either blow up or converge to zero? Well, it all depends on the initial value of x and in this example it is easy to understand just by looking at the numbers. You see, any number which is greater than the number 1 will get larger when you square it. It is as sure as the earth has gravity. Put the number 2 into a scientific calculator and hit the 'square' key. It doubles in size, right? After all, that is what multiplication by 2 means. Put 1.5 into a scientific calculator and it also gets larger, 2.25. Conversely, any number which is less than the number 1 will get smaller when you square it. Put 0.5 into a hand calculator and press the 'square' key you get 0.25, exactly half the previous value. That is what decimal 0.5 means, it is the decimal equivalent of $1/2$. You'll find that for any number which is less than 1, pressing the 'square' key will make it smaller.

Remember the joke about the chicken crossing the road? Well..., never mind.

So, the same equation can either blow up or converge to zero depending on its initial value of x .

WHY DO WE NEED THIS?

Guess what! The property that we test for when we want to produce a graphic image of the Mandelbrot set is the equation's convergence. Coincidentally, it also either converges to some low value (although not zero), or it blows up!

SAX MAN SAYS,

1 - AMIGA Computer
AND 1 - SaxMan Accounting System
or 1 - Billing & Disbursements System

and YOU'RE IN BUSINESS!!

SaxMan Systems has taken the best data base available for the AMIGA(tm)-SUPERBASE PROFESSIONAL(tm) from Precision Software, Ltd.-and created two very powerful and easy to use accounting systems.

Billing & Disbursements(tm), designed for the small business, processes cash and credit sales, prints invoices and statements, and ages accounts receivable. It tracks cash disbursements, prints a check register and reports on general ledger accounts sorted by category.

List Price \$249.95

SaxMan Accounting System(tm) is a completely integrated, multi-divisional accounting system with General Ledger as its hub. It handles Accounts Receivable, Accounts Payable, Payroll, Job Cost Analysis, Inventory Control, and a revolutionary new approach to payroll tax tables! Many easy to use features allow you to follow all data from source to final reporting in the General Ledger.

List Price \$499.95

Call: **SAXMAN SYSTEMS**
400 Walnut St., Suite 403
Redwood City, CA 94063
(415)368-6499

Circle 103 on Reader Service card.

To demonstrate this I will write a program which uses the simple equation from last month's article, except that this month I will write it, so that it uses convergence. If you have accepted, without complaint, that the equation $y = x * x$ converges to zero when x is less than 1, then you should have no trouble following this. By using the simple equation from last month, you will be able to concentrate on exactly how we perform the convergence test, without being distracted by any unfamiliar mathematics. You can run the program and confirm that a bunch of concentric circles are produced. Then I will do something exciting. I will replace the circle equation in the 'Calculate' subroutine with the Mandelbrot equation, without changing anything else. The program will operate exactly the same way, testing for convergence, but the pattern produced will change from simple circles to the much more complex Mandelbrot island. Like magic!

Our purpose this month is not to understand why the Mandelbrot set converges. One step at a time. Here we learn that to draw the Mandelbrot set we must test for convergence. And, we must feel comfortable programming the Amiga to do so.

START WITH A TEST FOR BLOWING UP

Let's go back to our example, $y = x * x$, which I now write like this:

$$x(n) \leftarrow x(n-1) * x(n-1)$$

This shows that we are going to be repetitively feeding the OUTPUT of the equation back to its INPUT. Let's pretend that we did not

AutoPrompt

by
DigiSoft

TV SCRIPT PROMPTER FOR THE AMIGA

AutoPrompt is a sophisticated scrolling prompter and text edit program designed for TV script prompting and information displays. It has been developed in consultation with TV industry professionals, with flexibility, simplicity and speed as the primary design considerations.

- AutoPrompt has a full intuition interface with menus and keyboard shortcuts.
- Selectable font, font size, pallet, speed and scroll direction.
- Markers for instant movement to pre-defined cue points.
- Message window and graphic scroll speed indicator.
- Simple easy-to-use menu driven text editor with load, save and print options.
- Imports ASCII or IFF FTXT format text files.
- Supports PAL and NTSC in both interlace and non-interlace, high or low resolution.
- Requires 512k of memory and Kickstart 1.2 or later.

AutoPrompt\$US 295.00
Demonstration disk...\$US 7.00

DigiSoft
12 Dinmore St
Moorooka
Brisbane 4105
Queensland Australia

International toll free order numbers from the USA and Canada:
from USA 1-800-525-2167
from Canada 1-800-663-3940
other countries 61-7-277-3255
within Australia (07) 277-3255

AutoPrompt is copyright (c) DigiSoft 1990
Kickstart is a trademark of Commodore-Amiga Inc.

Circle 137 on Reader Service card.

already know that this equation blew up for values of x greater than 1, and let's devise a method to test it.

It is easy to test if an equation blows up, just test if the equation ever exceeds some very high value. Let's choose the value 1000 and run the equation in a loop which tests it each time to see if it ever exceeds 1000. AmigaBASIC's 'WHILE-WEND' construction is very powerful for doing this.

```
x = 1.5
WHILE x < 1000
    x = x * x
    PRINT "X ="; x
WEND
PRINT "The equation blew up!"
```

The WHILE-WEND construction will repetitively execute everything between the WHILE and the WEND as long as the logic statement following the WHILE is true. The first time through the loop x equals 1.5. The logic statement $x < 1000$ is obviously true, so the calculation line within the loop gets executed. The result, 2.25, gets printed. Then the WEND statement sends the computer back up to the WHILE statement and again the logic statement $x < 1000$ is tested and found to be true. So, again the calculation line is executed, but this time with an INPUT of $x = 2.25$. The OUTPUT is 5.0625. It is growing. Well, we know from our previous discussion that the value will grow each time through the loop, and it will eventually exceed 1000. At that point the loop will terminate and the program will print "The equation blew up!"

There was nothing magic about choosing the value 1000 in my example. I could have picked any large value. If the equation is really going to blow up, then it will eventually exceed any number that I pick, no matter how large. I must however be careful to pick a value which is high enough so that the program does not incorrectly think that an equation is blowing up when it really is not. In this example, of course, we could have tested using a lower value like 5 or 10, but not a value less than 1.

TEST FOR CONVERGENCE

What happens in cases when the equation does not blow up? Like for input values less than the number 1 in our above example. In that case, the output of each iteration gets smaller and smaller and the expression $x < 1000$ is always true. The loop will continue to execute forever. The program will never end. We do not want that. We must therefore include a second test in our loop.

Note that although the above equation conveniently converges to zero, not all equations do this. Some converge to other values, like 1, or 5, or just about anything, and we want a test which will work for all these situations. Try this: test to see if the equation is not blowing up. Sound silly? Well... if it is not blowing up, then it must be either converging to or remaining less than some low value, right? An easy way to do this is to count the number of iterations in the same loop that we calculated before. We then decide that if after a certain number of iterations, say 500, the equation is still less than 1000, then it has probably converged to some lower value. The program below tests for both blowing up and converging, using the above idea.

```
FOR i = 1 TO 5

    INPUT "Enter a number"; x
    k = 0
    WHILE (x < 1000 AND k < 500)

        x = x * x
        k = k + 1

    WEND

    IF k = 500 THEN
        PRINT "The equation converged"
        PRINT
    ELSE
        PRINT "The equation blew up after "; k; " loops!"
        PRINT
    END IF

NEXT i
```

Note that the variable k counts the number of times that the loop is executed. Two things must be true for the loop to continue executing. First the value of x must be less than 1000, and second, the number of times that the loop is executed, k , must be less than 500. If either of these conditions ever become false, the loop terminates and the computer continues with whatever line follows the WEND statement.

WHAT'S HAPPENING?

To find out whether the equation blew up or converged to some value less than 1000, we test the value of k , the number of times that the loop was executed. If $k = 500$, that means that the value of the equation never exceeded 1000. We conclude then that the equation has converged to some lower value. If, on the other hand, k is less than 500, we conclude that the equation must have exceeded 1000; it blew up.

I have included some logic statements in the above program which demonstrate this. Run it several times using different initial values for x. You will find, quite properly, that whenever the initial value of x is less than 1, the equation executes 500 times and the program reports that convergence has occurred. Otherwise, the program reports that the equation has blown up after some number of iterations.

SPEED OF THE EQUATION

Oh no! Not another concept! Relax! This one is easy. Let's talk about initial values which are greater than 1. You may have noticed that for initial values which are close to the number 1, like 1.2 or 1.15, the value of k reported by the above program is greater than for initial values which are larger, like 10 or 25. When the initial value of x is very close to 1, it takes more iterations through the loop to determine that the equation is blowing up. We say in this case that it blows up slowly. On the other hand, a larger initial value requires only a few iterations to make that determination, and is said to blow up fast. Get the idea? The value of k is an indicator of the speed at which the equation is blowing up. The less the value of k, the faster the equation is blowing up. This concept will prove to be very valuable when creating different artistic effects with the Mandelbrot equation. But let's not get ahead of ourselves.

DRAWING THE CIRCLE

Remember how last month we used the horizontal and vertical coordinates of a pixel in an equation to produce a circle? We calculated the equation:

$$\text{Result} = i * i + j * j$$

Then we tested to see if 'Result' was less than 25. Perhaps you did not know it, but the right side of that equation represented the square of the radial distance of each pixel from the center of the screen. That fact comes to us through the centuries from a man called Pythagoras, a Greek who lived in about 550 BC. Let's go over this slowly.

We know that the coordinates of a point on a Cartesian plane, or any drawing surface, are a measure of its distance, both horizontally and vertically, from a zero reference point called the origin. A horizontal coordinate of -10 means ten units to the left of the origin. Similarly, a vertical coordinate of +7 means seven units above the origin. So, in last month's example the upper left hand corner of the screen was 10 units to the left and seven units above the origin. That's sort of like saying that your mother-in-law lives ten miles west and seven miles north of where you live. So, how far away is your mother-in-law, really? Well, it all depends on how you go. If you travel ten miles west, then pull a right on Misery Lane and go north for seven miles, you will arrive at your mother-in-law's place with seventeen miles wear and tear on your car. But what if you could travel in a straight line, directly in a north-west direction?

Years of geometric experience guarantee that the exact distance can be calculated by using the square root of the sum of the squares.

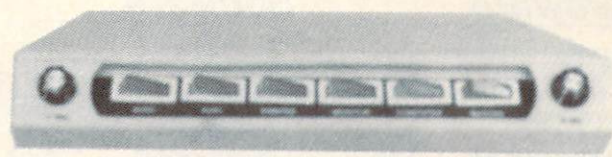
$$\text{Distance} = \text{SQR}(\text{west} * \text{west} + \text{north} * \text{north})$$

(SQR means square root)

This can be proved, but I will not do that here. Instead, I will simply trust Pythagoras. It is the old right angle triangle trick.

What about the OTHER half of MultiMedia?

AmiTech Computers presents...



AmiSoundTM

Sure, graphics are great. And we have, with little dispute, one of the BEST graphics machines around. The ability to directly manipulate video means that the Commodore Amiga will be, and is becoming THE major force to be reckoned with as the concept of MultiMedia comes into it's own (reading some of the "other guy's" magazines would lead you to believe that they invented the concept - the Amiga has had the capability for 5 years, only we called it Desktop Video!).

BUT as good as the graphics are, the OTHER side of the MultiMedia coin is often overlooked - high quality SOUND. The Amiga with it's direct stereo output is a start, but we here at AmiTech Computers have found a way to make it even better.

AmiTech Computers presents - AmiSound, the zero footprint solution for adding great sound to any Amiga. The internal 6 watt per channel stereo amp provides all the room-filling sound you need to get the most out of your desktop presentations (...and even your favorite arcade games!). The AmiSound also acts as a handy switchbox to put all your on-off switches up front - where you can reach them. Built-in surge suppression helps protect your valuable equipment from those dangerous line voltage spikes. This will be one of the most indispensable peripheral you will ever buy for your Amiga system! And now, take advantage of our NEW LOW PRICE! Call TODAY!

NEW LOW PRICE!

\$69.95



CALL or WRITE:

P.O. Box 65
Boystown, NE 68010
Phone: 402-493-7852



Circle 114 on Reader Service card.

Similarly, the direct distance between any pixel and the center of the screen is:

$$\text{Radius} = \text{SQR}(i * i + j * j)$$

Hence, last month we produced a circle of radius $\text{SQR}(25)$, or simply 5, by testing if 'Result' was less than 25. This month I also want to draw the same circle, but this time using the concept of convergence. Here's what I will do. First, I will calculate the square of the distance of each pixel from the origin, just like last month:

$$r = i * i + j * j$$

I used the letter r because it is the first letter in the word radius. The actual radius is the square root of r, but that does not really matter here. We can use the r value directly and still get the same pattern on the screen. Then, I will feed that number to the equation that we have been talking about in this article, $y = x^x$. I will simply replace the x's with r's.

$$r(n) <- r(n-1) * r(n-1)$$

Our experience tells us that if the initial value of 'r' is less than the number 1, the equation will converge to zero. On the other hand, when 'r' is greater than 1, the equation will blow up.

UNDERSTAND THE PLAN

The square of the distance of every pixel from the origin is first calculated and then sent as INPUT to our calculation machine.

**MicroMiga****Over 1300
Amiga Titles!****Hardware**

Amax Mac Emulator	\$128
Boling Mouse	\$99
CA-880 Floppy Drive	\$135
Internal Floppy	\$94
Deluxe MIDI	\$66
Digi-View Gold 4.0	\$135
Dual Serial Port Board	\$234
GVP A3001 Accel.	\$2,500
Kronos 500/2000	\$252/\$214
Magni Genlock System	\$1,684
Microbotics Hardframe	\$250
Microbotics 8-up Ram	\$151
Trump Card 500/2000	\$246/\$152
Quantum Prodrive 40s	\$497
ScanLab	\$815
Seagate Hard Drives	\$CALL
Sharp Scanners	\$CALL
Star NX-1000 MultiFort	\$193
SupraModem 2400	\$119
SupraRAM 500 (A501)	\$99

Software

Ami Alignment	\$33
Arexx	\$32
Bars & Pipes	\$192
Baud Bandit	\$32
DataStorm	\$26
Deluxe Paint III	\$105
Digi-Paint 3	\$65
Distant Suns	\$44
Falcon	\$33
GFA Basic	\$90
Lattice C v5.04	\$193
Lords Rising Sun	\$33
MusicX	\$175
PageStream v1.8	\$129
Pen Pal	\$88
ProWrite 2.5	\$80
Sculpt/Animate 4D Jr.	\$99
Shadow Of The Beast	\$30
Space Ace	\$35
Superback	\$53

1-800-733-AMIGA

FAX 619-670-9732 • PO Box 2104 La Mesa, CA 92044 • BBS 619-670-1095

At MicroMiga our customer and their Amiga are Number One. We carry over 1300 Amiga products, WAY too many to list here. Defective items repaired or replaced within 10 days of delivery. Non defects subject to 20% restocking fee. Shipping charges are \$4.00 first item and \$1.00 for each additional item. Hardware shipping rates by quote only. VISA & MasterCard accepted at no additional charge! Get the best for less!

Circle 182 on Reader Service card.

All the pixels that cause the equation to converge to zero will be colored black. All other pixels will cause the equation to blow up, and we will give them some other color depending on the above mentioned speed of expansion of the equation, the number of times through the loop, the k value. The complete program appears in Listing One.

Notice the use of the variable 'Crunch' to represent the maximum number of times I allow the WHILE-WEND loop to execute. It is initialized to 500 at the beginning of the program. Notice also the use of the variable 'M' to represent the maximum value that I allow the equation to grow. It is initialized at the beginning of the program to equal 4. We know by now that there is no need to allow the equation to grow as high as 1000 to determine if it is blowing up, 4 is sufficient. The 'calculate' subroutine ends with the number of iterations stored in the variable k.

CHOOSING THE COLORS

The actual color selection is done in the subroutine 'Select.Color' which is called immediately after completing the 'Calculate' subroutine. At this point the variable k is tested in order to make color decisions. If k = Crunch, I choose color register 2, which is black. Whereas, if k is some other value, a different color register is chosen - more about this aspect of designing fractals next month. Notice that in this program I did not use the usual 'Choose.Color.Numbers' and 'Use.New.Colors' subroutines like I did in previous months. I'm trying to keep things as simple as possible, so you can see and learn how to test for convergence.

The result of all this is that you will see a bunch of concentric circles of different colors. The one in the center will be black, and it represents all pixels which caused the equation to converge to zero. It is the solution set of the equation. The other circles will be of various colors, each one representing pixels which have caused the equation to blow up at different speeds. The larger circles represent the faster pixels, the smaller ones the slower pixels.

If all we wanted to do was draw a bunch of circles, the above method would be pretty inefficient. It is very slow. AmigaBASIC has some graphic commands that can do the job about ten billion times faster! But what we really want to do is draw fractals, and AmigaBASIC has no direct commands to do that.

NOW THE MANDELBROT SET

Replace the 'Calculate' subroutine with the one in Listing Two, keeping everything else the same. Although you cannot be expected at this point to understand the actual calculations involved within this subroutine, you can at least understand that an equation is being repetitively calculated and tested for convergence. The principle is the same as in our circle program. The WHILE-WEND loop executes until either k equals 500, the value of Crunch, or r exceeds 4, the value of M. For each pixel, the 'Calculate' subroutine ends with the number of iterations stored in the variable k. We then test the value of k to see what happened. If k = 500, we conclude that the equation has converged to some value less than the number 4. It would probably loop forever if we did not stop it at 500. We color it black. On the other hand, if the value of k is less than 500, we conclude that the equation has exceeded M, it blew up. We color it some other color depending on how fast it blew up. Almost like magic, the circles of your previous program are replaced by the more complex looking pattern of the Mandelbrot set.

OTHER POSSIBILITIES

Remember how I told you, in the first article, that with a little knowledge of the mathematics you could draw fractals that perhaps no one else has ever seen before? Listing Three presents a 'Calculate' subroutine which draws a solution set that looks like a distorted Mandelbrot set. Replace the 'Calculate' subroutine in Listing One with this modified one and see what happens.

Wow! A whole new world to explore. That's right, and anybody with a home computer can explore it. In fact, Figure One & Two show two fractals which were generated using this modified subroutine. Now you are probably thinking that you have to understand what the calculations of the Mandelbrot set mean, before you can make modifications like this yourself. It certainly would help. But in fact, even if you do not completely understand the math, there is nothing stopping you from introducing a few random modifications (mutations) and testing their results. You may accidentally stumble into a completely new world.

A PREVIEW OF WHAT'S TO COME

Next month we will go through the full design of a fractal, step by step. I will explain how you can create different artistic effects by making various color decisions.

[Please refer to pages 91 & 92 for the fractal listings.]

World Of Amiga New York '90



On April 27 and 28, The World Of Amiga, New York exposition attracted 15,233 attendees and over 40 companies. New York's Pier 92 was the setting for both new products and new companies to "set sail" on the Amiga market. Companies either attended WOA themselves or provided support for others—**Great Valley Products** was a major factor with a discrete 68030/50 MHz accelerated Amiga in the booth of New York Dealer **Amagination**.

Commodore—Front and Center

Attendees were greeted by **Commodore Business Machines** new booth which sported a host of Amigas and activities. Commodore used WOA to demonstrate the **Amiga 3000** and their new, still unreleased authoring system, **AmigaVision**.

AmigaVision is a new software package to be bundled with all new A2000 and A3000 systems. Driven by an icon interface, **AmigaVision** permits users to create complicated presentations in a flowchart format without resorting to a scripting language. Users can manipulate text, graphics, Amiga IFF pictures, animations, digitized sound, speech synthesis, Videodisc control of Sony & Pioneer players, and more. While the package allows preloading of audio and graphics to increase speed, it also manipulates external programs through **ARexx™**. For Amiga owners who did not purchase their machine within ninety days of the **AmigaVision** announcement, CBM will be offering **AmigaVision** for \$149. CBM's apparent aim is to entice every Amiga owner to create presentations quickly and effortlessly on the Amiga.

Everyone wanted a firsthand look at the new Amiga 3000. With its advanced technology, superior speed, great looks, and low price tag, the A3000 provides dramatic competition to Apple's Mac. (For a complete description of the new Amiga 3000, please see the coverage in AC's May issue.)

CBM backed up its "New Commodore" commitment with open speeches and forums provided by Commodore top executives. Both Lloyd Mahaffey, Director of Marketing, and Harold Copperman, President and CEO of CBM, spoke before large audiences.

During Gold Disk's presentation of **Showmaker**, Mr. Mahaffey delivered the message that Commodore is aiming directly at the multimedia market. He stressed the need for products such as **Showmaker** and **AmigaVision** to create advanced multimedia presentations with the Amiga—presentations which cannot currently be produced on any other computer platform.

Mr. Copperman announced that Commodore would now warranty all Amigas for one full year from the date of purchase. This announcement covers all Amigas currently under their ninety-day warranty. Mr. Copperman also announced a free pick-up and repair service through Federal Express for Amiga 500 computers under warranty. Mr. Copperman was enthusiastic about the recent press Commodore has received and the new products and management teams now in place at CBM.

Gold Disk's Showmaker Takes A Bow

Gold Disk started festivities at 10 AM on Friday with the first live presentation of

their new multimedia presentation and designer package, **Showmaker**. Gold Disk President, Kailash Ambwani, stated: "Showmaker fills a critical need in multimedia and desktop video. Much as a conductor of an orchestra is the person who is responsible for bringing in all the different components of an orchestra, making sure they all work together in harmony, making sure they are all synced and meet their marks at the right time, that is exactly what **Showmaker** does. **Showmaker** integrates all the different mediums—sound, graphics, animation, live video, music, titling—and takes all of these components and integrates them together into a multimedia presentation or a video."

Gold Disk presented **Showmaker** as the first program with intelligent autoloading of animations and graphics, animation playback synchronized to music in frames-per-beat, background music looping for easy creation of sampled soundtracks, complete software control of genlock functions, video titling over Amiga animation and graphics, production looping, and MIDI sound effect support for professional effects.

Showmaker's graphic interface allows anyone to modify their presentation by simply dragging out the time bar for that sequence. The bar can refer to music, animation, sound effects, text, video disk input, or genlock activity. **Showmaker** permits external clock input from MIDI or MTC. Utilizing the Amiga's multitasking capability, **Showmaker** will load files for future needs while the present animation is in progress. This allows the animation, presentation, or video editing process to continue undisturbed.



One of the main advantages of using a system like Showmaker lies in your ability to create a fully integrated video or multimedia presentation, edit it repeatedly, and then take the entire collection of components to a video lab to produce a master tape in one take. Showmaker presentations are only limited by the size of your hard drive and the other external sources such as laserdisks and CD players.

Gold Disk also announced **Professional Draw 2.0** would be available this month. The \$295 package is five times faster than previous versions and contains an auto-tracing feature which will generate objects by automatically tracing over bitmapped images. Professional Draw now will write text along curves, permit color dithering (for an effect of over 1000 apparent colors on the screen at once), and "blend" any two lines, colors, or objects.

Gold Disk noted their new **Gold Disk Type** series will consist of packages of three compugraphic fonts per pack (\$59.95). The **Outline Fonts** is a collection of over 35 Compugraphic outline fonts (\$199.95). All of these resizeable fonts can be used in any program that accepts Amiga fonts. They are best known for their use in desktop publishing, but are just as handy in video titling, word processing, and graphics.

Pre'spect Technics Inc.

Thomas Bothe and Horst Bothe were extremely busy as they demonstrated several new products they had just brought in from Canada. Long known for their A.L.F. (Amiga Loads Faster) controllers for the A500, 1000, 2000, MFM/RLI ST506 or SCSI, Pre'spect introduced six new additions. These included a **Non-Flicker Cable**, an **Amiga/VGA Switch Cable**, a **TTL Monitor Cable**, a backup program for hard disks and changeable disks, the **FileRunner Streamer**, and a new feature in their **ALF2** software which allows the use of AMax with any hard disk.

In the cable area, Pre'spect now offers inexpensive solutions for expensive situations. Their Non-Flicker Cable is a \$38.50 cable that attaches an Amiga to a TTL Hercules monitor for flicker-free performance. The TTL monitor cable is designed to attach a full-page monitor to an Amiga. The Amiga/VGA Switch Cable was created for people who use a bridgeboard, a multisync monitor, and a VGA card. They can switch between the Amiga display and the PC display at the flick of a switch.

Free Spirit Software Plays Doctor

Recently, **Free Spirit Software** has received recognition for their educational game, **Barney Bear Goes To School**, and the arcade-style fantasy, **Dragonscape**. Yet they also publish a program to calibrate and correct Amiga floppy drives, **Ami Alignment System**. In this tradition, they have just released a new Amiga program called **Doctor Ami**. Consisting of two programs (Drive Doctor and Memory Doctor), Doctor Ami is a memory and hard drive diagnostic program designed to test memory and hard drive sectors. Memory Doctor scans all system and expansion memory for any bad sectors and then maps them out. Drive Doctor reads all disk sectors, automatically mapping out any bad sectors. Doctor Ami retails for \$49.95.

Pulsar Creates A PC-Compatible A500

Pulsar is the creation of long-time Amiga enthusiast Eric Miller. Most Amiga users know Eric and his alter ego, Dr. Oxide, from the Amiga retailer CompuSave. Eric dropped his Dr. Oxide attire for a suit on Friday in order to present Pulsar to the public. Besides incorporating current products from companies such as **Joe's First Company** and **Micro-Momentum, Inc.**, Pulsar wowed A500 users with a new expansion card for the Amiga that makes the Amiga PC-compatible.

The **POWER PC BOARD** fits in the expansion slot beneath the A500 and is both an expansion card for the A500 and a PC-XT computer. With the addition of this card, it is possible for A500 users to run MS-DOS software directly from their internal 3.5" drive, or it will support external 3.5" and 5.25" drives. **POWER PC BOARD** is designed to support Hercules/CGA and Monochrome video displays. The hardware includes Phoenix Bios, Clock/Calendar, 704K RAM plus 64K EMS and a NEC V30 8MHZ. Software includes Microsoft 4.01, MS DOS Shell, GW Basic, and, on the Amiga, CrossDOS. Pulsar also supplies a spreadsheet, database, and word processor for the PC with each unit.

POWER PC BOARD supplies the Amiga with an additional 512K RAM as well as a 512K RAM Disk. According to the demonstration, the **POWER PC BOARD** runs faster than a comparable PC or XT, and even faster than Commodore's bridgeboard for the A2000. The **POWER PC BOARD** also takes advantage of the serial port, parallel port, joystick port, and mouse. Pulsar was offering the entire **POWER PC BOARD** package at an introductory price of \$550.00.

Soft-Logik, PageStream 2.0 and More!

Soft-Logik Publishing Corp., the St. Louis, Missouri developer of PageStream, used World Of Amiga to announce **PageStream 2.0**. This is an upgrade to their PageStream 1.8 released in January 1990 (please see the review on page 14 of this issue).

Soft-Logik's President, Deron Kazmaier, and company have been extremely busy. Along with the release of PageStream 2.0 scheduled for this month, their press package distributed at WOA included announcements on licensing agreements with AGFA-Compugraphic and Image Club Graphics, Inc., a new co-operative relationship with LaserUp!

Software in San Francisco, a special user group promotion for PageStream 2.0, as well as a new addition to their line, **Business Forms**.

Business Forms is a collection of 40 customizable forms for day-to-day use in accounting, invoices, office memos, and purchase orders.

Licensing AGFA-Compugraphic's Bullet outline font technology allows Soft-Logik to incorporate new fonts into PageStream 2.0. Some of the new fonts available under this agreement are CS Times in Roman, Bold, Italic, and Bold Italic as well as CS-Triumvirate and CS-Triumvirate-Bold. These hinted outline fonts allows PageStream to produce quality fonts on the screen and output to dot-matrix printers.

The additional licensing of Image Club Graphics, Inc.'s **Image Club** collection will allow Soft-Logik to develop and incorporate this large variety of clip art now available for the Macintosh into the Amiga market. The current library contains over 2000 Encapsulated PostScript (EPS) images. These images can be reduced, enlarged, or distorted without a loss in reduction. The type library, Adobe Type 1 compatible, contains over 600 licensed typefaces.

PageStream 2.0 appears to have all the features of V1.8 plus additional capabilities provided by the new agreements mentioned above. It utilizes the AGFA-Compugraphic hinted outline fonts as well as the Adobe Type 1 IBM format (without hints). In addition, PageStream 2.0 allows users the precision of 1/100 degree increments in rotation of objects and the ending angles of arcs. Font sizes can be specified in 1/100th point increments and the position or size of an object can be set to 1/100th point increments as well. Registered owners of previous versions of PageStream will be able to upgrade to 2.0 for \$75.00.

User groups can, by written request, receive two copies of PageStream as well as a demo copy for their library. It is hoped that one copy will be used for newsletter reviews and production, while the second is used as a prize by the group. User groups are also eligible for group discounts through their dealers.

NewTek's Toaster Meets The Amiga Community

NewTek demonstrated its full working **Video Toaster** before packed audiences throughout WOA. If Amiga enthusiasts were not standing in awe of the special effects and graphic capabilities displayed on the large overhead monitors, they were watching the continuous showing of NewTek's Penn & Teller demonstration tape for the Video Toaster.

The Toaster's brochure promises a multitude of features including a character

generator, digital video effects, color processing, dual frame buffers, production switcher, preview output, linear keyer, digital still store, a real-time (1/60th of a second) frame grabber, integrated design and more. All of these features are available through an intuitive interface. Each process of the Toaster is accessible with the Amiga's mouse.

The heart and soul of the Toaster is its ability to control four video inputs and produce hundreds of digital video effects. All of the hundreds of different digital effects can either be manipulated with the clear graphic interface or by using the keyboard. The color processing engine can create color negatives, monochrome, solarization, posterization, color vignettes, and photographic filter effects.

The Toaster generates 35 ns text in 16.8 million colors with variable shadows and other effects through its built-in character generator. The character generator comes with 25 standard fonts, plus it permits unlimited user-definable fonts. With 100 pages of on-line storage, graphic separators, full spectrum background color graduations, the Toaster even offers a variable speed smooth crawl and scroll at 60 fps.

The Video Toaster's highly professional output device was praised by graphic artists for its two 24-bit, broadcast-quality frame buffers. Not only does this allow the Toaster to hold two separate digital images, but, with the correct software, it enables 24-bit graphic rendering and painting in these two areas. Although NewTek has adhered to the 24-bit graphic IFF standard from Commodore, none of the major 3D, painting, or ray-tracing products currently available will render to these requirements without using a conversion program. (Several such programs are currently available in the public domain, according to a NewTek executive.)

NewTek is now promising early shipments of the Toaster in late June with full quantities available for dealers in July. Citing FCC requirements and testing as the reason for the delay, a NewTek spokesperson was very confident that



NewTek will be able to supply the entire Amiga market quickly. Company executives have stated that the demand for the Toaster has skyrocketed following its demonstration and success at The National Association of Broadcasters convention in April. (Please see the related article in the May issue of *Amazing Computing*).

Notwithstanding the stir created in professional markets, NewTek remains confident that the Toaster will be used effectively by a great many non-professional Amiga users. While commercial equipment currently does not give the casual videophile access to tools which provide completely professional results, a spokesperson for NewTek was very excited by the use of the Toaster in live video work.

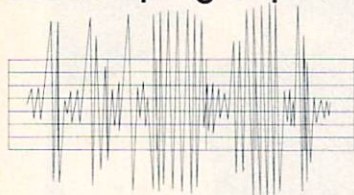
With luminance keying, frame grabbing, two 24-bit cards, character generation, and countless digital effects all manipulated by mouse or keyboard on a graphic interface, NewTek has aimed for a wide market—anyone who wants to do more with video. The Toaster's price has remained a very attractive \$1595, designed to appeal to the home user, graphic artist, and television producer alike.

ProWrite 3.0

New Horizons Software, Inc. distributed their first copies of **ProWrite 3.0** (\$175). Among the many features of the

AudioLink

**16-bit Linear Stereo
Audio Processor with
Sound Sampling Capabilities**



Beta Unlimited
87 Summit St. Brooklyn, NY
11231

Circle 126 on Reader Service card.

new ProWrite is a modified page layout technique that, when added to its graphics capabilities, is a boon to video scripting. With the addition of ProScript (\$49.95), ProWrite will print to any Postscript printer. New Horizons has provided a page of system features from better print utilities to the best feature of all—a Bug Free Guarantee. New Horizons has been applauded enthusiastically for marketing a product that they guarantee will be trouble-free. They have established a new level of professional quality for Amiga products.

Migraph Hand Scanner and Touch•Up™

Migraph demonstrated their new entry into the Amiga market, the **Migraph Hand Scanner and Touch•Up™** software. The hand scanner is over 4 inches

wide and allows the user to scan images by hand into a black-and-white format. Amiga owners with 1MB of memory (and no other applications) can scan in a 4" by 8" image in 300 dpi, while Amiga users with 2MB of memory can scan an area up to 14" tall at 400 dpi.

The Touch•Up™ software allows users to manipulate their scanned images or other images imported from other formats including: IFF, IMG, PCX, TIFF, MacPaint™, and GIF. The Scanner was sold with a .9 pre-release version of the software at the show, but finished versions will be sent to all users when the product ships this month. The scanner with software will retail for \$399.

(continued on page 95)

World Of Amiga, New York '90 Exhibitors

Abacus Software
5370 52nd St. SE
Grand Rapids, MI 49512
(616) 698-0330
FAX (616) 698-0325
Inquiry #214

Accolade
550 S. Winchester Blvd. #200
San Jose, CA 95128
(408) 985-1700
FAX (408) 246-0885
Inquiry #215

Amagimation
130 W. 26th St.
New York, NY 10001
(212) 727-3290
FAX (212) 727-3293
Inquiry #216

Amiga Video Magazine
200 W. 72nd St. Suite #53
New York, NY 10023
(212) 724-0288
Inquiry #218

Beta Unlimited
87 Summit St
Brooklyn, NY 11231
(718) 852-8646
FAX (718) 624-5738
Inquiry #219

Britannica Software
345 Fourth St
San Francisco, CA 94107
(415) 597-5553
FAX (415) 546-0153
Inquiry #220

Briwall
P.O. Box 129
58 Noble St.
Kutztown, PA 19530
(215) 683-5433
Inquiry #221

Broderbund Software
2888 Bluff #546
Boulder, CO 80301
(303) 499-3266
Inquiry #222

California Freeware
1747 E. Avenue Q. C-1
Palmdale, CA 93550
(805) 273-0300
FAX (805) 273-0672
Inquiry #223

**Commodore Business
Machines**
1200 Wilson Dr.
West Chester, PA 19380
(215) 431-9100
FAX (215) 431-9156
Inquiry #224

Comp-U-Save
414 Maple Ave.
Westbury, NY 11590
(516) 997-6707
FAX (516) 334-3091
Inquiry #225

Data East USA
1850 Little Orchard Dr.
San Jose, CA 95125
(408) 286-7080
FAX (408) 286-2071
Inquiry #226

Disks & Labels to Go
Rt. East Hampton Business
Park
Mount Holly, NJ 08060
(609) 265-1500
FAX (609) 265-0818
Inquiry #227

Dr. T's Music Software Inc.
220 Boylston St Suite 206
Chestnut Hill, MA 02167
(617) 244-6954
FAX (617) 244-5243
Inquiry #228

**Electronic Arts/Lucasfilm/
Miles Computing**
1820 Gateway Dr.
San Mateo, CA 94404
(415) 571-7171
FAX (415) 571-7995
Inquiry #229

Electronic Zoo
3431-A Benson Ave
Baltimore, MD 21227
(301) 646-5031
FAX (301) 646-7753
Inquiry #230

Expert Services
5912 Centennial Circle
Florence, KY 41042
(606) 371-9690
Inquiry #231

Free Spirit Software Inc.
PO Box 128
58 Noble St.
Kutztown, PA 19530
(215) 683-5609
FAX (215) 683-8567
Inquiry #232

Gold Disk Inc.
2175 Dunwin Dr #6
Mississauga, Ont.
Canada, L5L 1X2
(416) 828-0913
FAX (416) 828-7754
Inquiry #233

Great Valley Products
225 Plank Ave
Paoli, PA 19301
(215) 889-9411
FAX (215) 889-9416
BBS (215) 889-4994
Inquiry #234

Hillside Hard & Soft
Mt. Laurel Rd.
RD3, Box 3420A
Fleetwood, PA 19522
(215) 929-9695
Inquiry #235

Hologramophone Research
6225 SW 145 St.
Miami, FL 33158
(305) 252-2661
Inquiry #236

Innerprise Software Inc.
128 Cockeysville Rd.
Hunt Valley, MD 21030
(301) 785-2266
FAX (301) 785-2268
Inquiry #237

INOVAtronic, Inc.
8499 Greenville Ave.
Suite 209B
Dallas, TX 75231
(214) 340-4991
Inquiry #238

**Media Tech/ Bethesda
Softworks**
15235 Shady Grove Rd.
#100
Rockville, MD 20850
(301) 963-2000
FAX (301) 926-8010
Inquiry #240

MichTron/MicroDeal
3285 Lapeer Rd W
Auburn Hills, MI 48057
(313) 377-8898
Inquiry #241

MicroProse
180 Lakefront Dr.
Hunt Valley, MD 21030
(301) 771-1151
FAX (301) 771-1174
Inquiry #242

Migraph Inc.
200 S. 333rd., Suite 220
Federal Way, WA 98003
(206) 838-4677
FAX (206) 838-4702
Inquiry #243

New Horizons Software
P.O. Box 43167
Austin, TX 78745
(512) 328-6650
FAX (512) 328-1925
Inquiry #245

NewTek, Inc.
115 W. Crane St.
Topeka, KS 66603
(913) 354-1146
FAX (913) 354-1584
Inquiry #246

Octree Software
311 W. 43rd St. #904
New York, NY 10036
(212) 262-3116
FAX (212) 581-7703
Inquiry #247

Precision Inc.
8404 Sterling St #A
Irving, TX 75063
(214) 929-4888
FAX (214) 929-1655
Inquiry #248

Pre'spect Technics Inc.
1085 St. Alexander,
Suite 500
Montreal, Que
Canada, H2Z 1P4
(514) 954-1483
FAX (514) 876-2869
Inquiry #249

Psygnosis Limited
Port of Liverpool
1st Floor, Pier Head
Liverpool, England L3 1BY
011 44 51709 6466
Inquiry #250

Saxon Industries
14 Rockcress Gardens
Nepean, Ont
Canada K2G 5A8
(613) 228-8043
FAX (613) 563-9596
Inquiry #251

Semitech Inc.
146 S. Lakeview Dr.
Gibbsboro, NJ 08026
(609) 346-9814
Inquiry #252

Sierra On-Line Inc.
P.O. Box 485
Coarsegold, CA 93614
(209) 683-3472
FAX (209) 683-3633
Inquiry #253

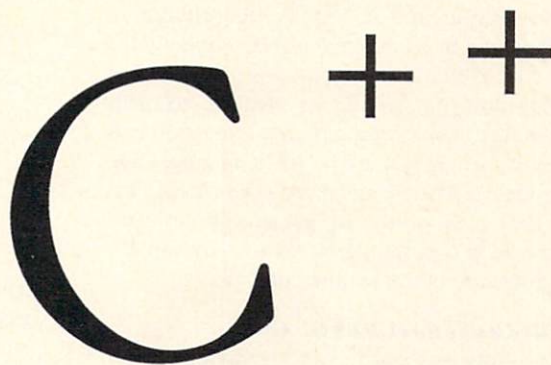
Soft-Logik Publishing Corp.
11131 F South Towne Sq.
St. Louis, MO 63123
(314) 894-8608
FAX (314) 894-3280
Inquiry #254

Software Link Inc.
197 E. Post Rd.
White Plains, NY 10601
(914) 683-2512
FAX (914) 683-5158
Inquiry #255

Spectrum HoloByte
2061 Challenger Dr.
Alameda, CA 94501
(415) 522-3584
Inquiry #256

Spirit Technology
220 W. 2950 S
Salt Lake City, UT 84115
(801) 485-4233
FAX (801) 485-6957
Inquiry #257

Zuma Group Inc.
6733 N. Black Canyon Hwy.
Phoenix, AZ 85015
(602) 246-4238
FAX (602) 246-6708
Inquiry #260



An Introduction to Object-Oriented Amiga Programming

by Scott B. Steinman, O.D., Ph.D.

INTRODUCTION

Look at just about any computer journal these days and you will see screaming from its pages “OOPs”! Is this a warning from software developers? In a sense, yes. But it doesn’t mean that your software may break – just the opposite. OOP is the abbreviation for Object-Oriented Programming, and what object-oriented programming promises is the construction of large programs that are easier to debug, maintain and modify. Rather than treating programs as a series of procedures executed one after another, object-oriented programs focus upon objects, which are groupings of data and allowed actions that act as descriptions of real-world entities and how they behave. Programmers learn to think in terms of the data itself and the problems they are solving rather than a particular

process of modifying the data. These descriptions are reusable, and once created, they can be “plugged into” other programs and adapted to new applications.

Several object-oriented languages have sprung up suddenly. Some adhere strictly to object-oriented concepts, such as SmallTalk, but these are also interpreted languages (like BASIC) and so are slow. Others are “hybrid” languages – object-oriented but built upon existing fast compiled languages. C++ is one of these hybrid languages. It is an extension of the C programming language that supports all the features of C, plus a lot more. C++ was designed by Bjarne Stroustrup of AT&T to be the successor to the C programming language – what he calls “a better C”. Let’s look first at what makes C++ better than C. ➔

“C++ offers more than enough new features to justify learning it.”

“Object-oriented programming is a new way to break down programs into small, easy to understand, reusable and adaptable modules.”

WHY BOTHER?

Learning a new programming language always requires effort. That learning effort must be justified by a later increase in programming ease once the new language is grasped. C++ offers more than enough new features to justify learning it. The strength of C++ is that you may start by using just a few of its features, and gradually add more and more as you learn, until you are truly producing elegant reusable object-oriented programs. We'll start by summarizing some of the features of C++.

NON-OOP FEATURES OF C++

Not every feature of C++ was added with object-oriented programming in mind. Some were added just to make C programs easier to write, more bug-free and more understandable. Let's look at a few of them.

C++ supports strong data type checking, which helps prevent you from using the wrong data type by mistake. We've seen this in C when we specify the argument types and return type of a function. However, in C, if you pass the wrong data type to a function, the program will continue to run, but run incorrectly. In C++, you write a function prototype that contains the argument types and return type before you use the function. The compiler will then generate an error message, if you supply the wrong data types when calling a function, giving you the opportunity to find your error before it finds you!

C++ allows you to create new data types. While C allowed you to do something like this with typedef, these “new” data types were simply the same old data types with new names, just like aliased command names. New data types in C++ really are new data types and are subject to type checking when they are used as function arguments or return values.

The new keyword is 'const'. It provides a way to define constants without using the preprocessor command #define. It avoids the errors that stem from preprocessor text substitution and allows strong type checking on constants as well. If a function makes one of its arguments a constant, the passed value is immune to changes. This allows us to pass pointers to functions without worrying about accidentally altering the data values pointed to.

Another new keyword is 'inline'. When we declare a small function to be inline, its code is substituted directly when the function is called, much like a macro definition, avoiding the overhead of calling a function. However,

inline functions are subject to argument type checking, unlike macros.

References, accessed with the & operator, permit the passing of function arguments by reference without the need to use pointer notation within the function. For example, if a function argument is int& x, we may access x without the *x notation even though we have passed it by reference rather than by value. This makes our code easier to read and understand.

C++ has additional memory allocation operators. Rather than call malloc() and free() as in C, we use the operators new and delete in a simpler notation. For example, to allocate a character string array str of 20 characters, we write str = new char[20], and to free it we write delete str.

Function arguments may now be supplied with default values. If the user fails to provide a value for an argument when calling the function, the default values are used.

Function overloading is a means of allowing several related functions with the same name to be called with different argument data types. Let's say we'd like to have a function called print() that we wish to call with either a character or a double argument. In C, we'd need to have two distinct functions, cprint() and dprint(). In C++, we'd need only to call a function with the single name print() with either a char or a double argument!

The input/output functions have been changed in C++. If we choose, we can avoid using the formatted I/O functions scanf() and printf(). It's too easy to accidentally try to print a double variable with %d, resulting in bizarre output. The new stream class permits you to do stream I/O in a more natural way that avoids errors. The stream class can also automatically recognize the data type of the variables you wish to print!

There are numerous other fine points of C++ that make programming easier. However, space prevents me from listing all of them here. Rather than fill up this article with a list of non-object-oriented C++ features, let's look at the more revolutionary world of object-oriented programming.

A GENTLE LOOK AT OBJECT-ORIENTED DESIGN

Before we examine how C++ supports object-oriented programming, let's talk a bit more about what object-oriented design is and what makes it so beneficial.

The key concepts of object-oriented programming are data abstraction, data encapsulation, data hiding, the class interface, inheritance and polymorphism. These are big words, but they really mean simple things. Simple but powerful things!

As computer programs get larger and larger, they also get harder to maintain. It's one thing to write a big program, but that takes only a small fraction of your programming effort. Let's face it – what really takes up the most time is debugging and modifying programs! Object-oriented programming is a new way to break down programs into small, easy to understand, reusable and adaptable modules. These modules are not C functions, but a unique combination of data and functions called a class. A class is simply a way to represent a real-world concept. Let's look at a trivial example: Let's say you need to write a program that simulates different models of cars. First, you would think about what parts make up a car – an engine, tires, a transmission, etc. – and what a car does – steer, drive and stop. All cars are built from common parts that make all cars perform similarly. Here's our object – the car. A car is characterized by both its parts and the actions it can perform, and our class car is similarly composed of car parts (data) and a set of car actions (functions or methods).

An object is a particular instance of a class. Each object shares the class' common data and functions, but may differ in the exact values of each data variable. In our example, a Ferrari object and a Porsche object are both instances of the car class, and so share the common features and behavior of all cars (that is, they both have engines and they both drive). However, the parts they are built from may differ slightly (such as different types of engines, different transmissions, etc.), even though they are still both cars. Similarly, objects created from the same class share common data variables and allowed actions, but may differ in their data's values.

Classes and objects provide us with data abstraction – the creation of new data types that represent real-world entities. The new data type (class car) can be used as if it were a built-in data type like an int or float. Data encapsulation is the term for the packaging of data and functions into classes. This packaging allows us to locate bugs more easily as well. If a program using a car doesn't work correctly because the data within a car object acts strangely, we know that the bug

must be within the car class' code only and not elsewhere in the program. Data encapsulation also allows us a new degree of control over our data. Control that helps prevent unwanted changes to the data. We may allow users of the class (yourself or other programmers using the class) access to certain class data and actions and not to others (data hiding). For example, we may allow a user of class car to drive it or park it, but not to directly control the engine timing or to remodel its parts! The subset of actions that users are allowed to do forms the class interface, the way that the programmer must use the class. It prevents the user's access to sensitive data or functions that we don't want them to use in ways we did not plan. If we don't specifically allow them this access, they don't have it! This helps ensure the proper use of the class – that is, that the user makes a car do only what we think a car should do.

The user interface and data encapsulation have a more important side effect – the ability to change our class' implementation details without affecting the program that uses the class! As an example, suppose you need to use sorted data. In C, changing how the sorted data is stored and the particular sorting function would require changes in all code that accesses the sorted data – quite a task in large programs. In C++, so long as the user of our sorted-data class accesses that data via the class interface, we can change the class' data representation and function algorithms as much as we want without the user's code being affected. All they must do is recompile and relink with our new class – no reprogramming is required!

Although classes may be used like built-in data types, they are far more useful. Classes that are written once may, if written correctly, be used over and over again in other programs without rewriting the same source code by placing them in class libraries. More importantly, they may be built upon and enhanced to adapt to new uses. This feature is called inheritance. We create new subclasses from our original class and have the option of adding new data and actions to it that extend its usefulness. The subclass shares all the properties of its parent class by inheriting them, and we may choose what properties we need to change for a given application. Returning to our car example, we can create different subtypes of cars, racing cars and street cars. Each shares the features of car, such as an engine and the ability to drive, but racing cars also have roll

➔

CANVAS

For the Amiga

This is a three disk collection of 13 animation demos and 5 pictures that you can load into your favorite animation editor, such as Deluxe Paint III. This collection of animations was developed in the style of traditional animation. One meg. of memory is suggested as these animations run at 15fps and range from 60 to 120 frames long.

Price: \$30.00 (+\$2.00 shipping)

LUNAR

Construction Disks

Create your own fantastic scenes of lunar landscapes, tumbling asteroids, and sparkling stars on the Amiga with these high quality, full color images. This 2 disk set contains over 100 pictures, brushes, and anim brushes -- your only limitation will be your imagination.

Price: \$25.00 (+\$2.00 shipping)

To order CANVAS or LUNAR Construction disk, please send a check or money order to:

Silver Fox Software

P.O. Box 551413

Dallas, Tx. 75355-1413

Call (214) 349-1681 for information and Dealer inquiries.

Circle 105 on Reader Service card.

bars and may race, while street cars have chrome trim and air conditioning and may parallel park.

Inheritance allows us to create general-purpose classes whose uses may be extended or adapted by creating more specific subclasses. As a more realistic example, we could make a general window class, a rectangular portion of the screen, and this may be carried further to make the subclasses textwindow for the presentation of text and picturewindow for the presentation of graphics. The time-saving benefit of inheritance is that you don't have to start from scratch to write programs with textwindows or picturewindows. Once the general data and algorithms of parent class window are debugged, they may be reused or modified by its subclasses textwindow and picturewindow, and only the changed features coded.

The last advance of OOP is polymorphism. This is the ability to have the system recognize at run-time which function to call! This makes programs more flexible. Rather than having to hard-code the specific function you're going to call before compiling a program, you may request a general action, and the system will know as the program runs which function to call. For example, let's

say we have a class point with derived classes rectangle and circle, with functions called draw() in each. We may then make a list of graphics objects to draw. When we draw() each object in the list in turn, the system knows whether to draw a point, rectangle or circle.

OBJECT-ORIENTED FEATURE OF C++

C++ supports all the key concepts of object-oriented programming, and does so with little sacrifice of program speed.

Due to the limited space for this article, we cannot look at all the ways that C++ supports object-oriented programming. However, just to whet your appetite, let me show you how easy it is to create and use a simple class, and in the process demonstrate a few other powerful features of C++ classes. Here's the definition of a class box in the header file box.h:

```
class box {
private:
    int x;
    double y;
    char *name;
public:
    box( int a = 6, double b = 1.2, char *boxname );
    ~box();
    void fillx( int a ) { x = a }
    void display();
};
```

The keyword class tells the C++ compiler that a class definition is to follow. Notice that a class looks a lot like a struct with functions in it as well as data. This class has two regions -- a private region and a public region. The private region may not be directly accessed by users of the class. It may only be accessed by public functions. In the class box, users cannot set the double y, since there's no fill y() function in the public region that could access y. We have protected our y variable from being altered by the user. The public region therefore forms our class interface -- the actions we allow the user to perform. Our class interface contains the functions fill x() and display(). As an aside, note that the functions are defined as in ANSI C (function prototypes) and allow the setting of default arguments that are used if the user does not call the function with specific values for the arguments. In addition, the member function fill x() has been supplied as an inline function (the keyword inline isn't necessary inside a class if the function code is written inside the class definition).

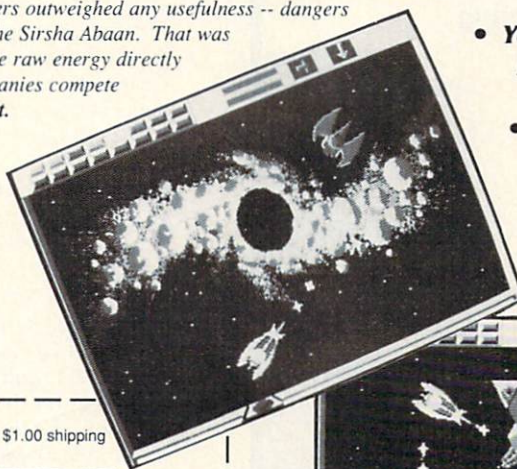
What is that oddball function box() in the class box? Why doesn't it have a return type? Why does this function's name match the class' name? This is a special function called a constructor. It's used for initializing the class, if we want to do so. In other words, we could have the data in objects of the class box initialized automatically when the objects are declared! This can include

Twilight Games Introduces NEW

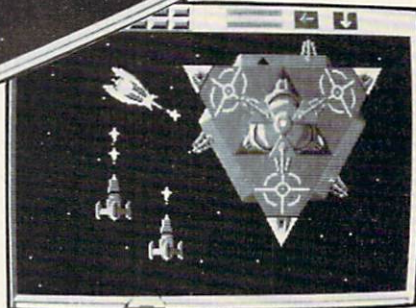
Corporate Raiders

52 years ago, the Orvo first explored what is now known as the Orvan Quadrant. The area was forgotten because the natural dangers outweighed any usefulness -- dangers like Gravity Wells, Asteroid Belts, and the Sirsha Abaan. That was before technology was developed to mine raw energy directly from the Gravity Wells. Now four companies compete for mining rights in the Orvan Quadrant.

- **Plan your Strategy!**
Then grab the joystick and take control
- **A perfect blend of Strategy and Arcade Action**



- **You control Multiple Patrols**
- **Form Alliances**
- **Obliterate Enemy Bases**
- **Plunder Enemy Grav Wells**



Please send me **Corporate Raiders**.
Enclosed is a check or money order for \$39.95 + \$1.00 shipping

Name _____
Address _____
City, ST Zip _____
Mail to: **Twilight Games** Or Call: **(503) 293-6593**
P.O. Box 273 COD and 2nd Day delivery available
Beaverton, OR 97075

Circle 175 on Reader Service card.

storage memory allocation for data arrays as well. The even stranger function starting with a tilde (~) is a destructor. It does the opposite of a constructor. When a class is no longer being used, the destructor provides automatic cleanup, so you don't need to remember to free up memory yourself.

How do we use the class? This is simple also. We create a file `box.cp` that contains the member function code:

```
#include <stream.h>
#include "box.h"
```

```
box::box( int a, double b, char *boxname )
{
    x = a;
    y = b;
    name = new char[ 30 ];
    strcpy( name, boxname );
}
```

```
box::~~box()
{
    delete name;
}
```

```
}

void box::display()
{
    cout << "Box: n" <<
    << "Name: " << name << " n"
    << "Contents: " << "x= " << x
    << "y= " << y;
}
```

We include the file `box.h` containing the class definition at the top of the `box.cp` file. The first member function we write is the constructor `box()`. It sets the values of `box`'s data members using the memory allocation operator `new` to create storage for the character string `name`. Note that member functions can access data members directly. The next function that we write is the destructor `~box()`. It frees the memory allocated in the constructor for `name` by using the `delete` operator. The final member function simply prints out the contents of the box. It takes advantage of the stream class for simple output to the standard output stream `cout`. The data types for `x` (`int`), `y` (`double`) and `name` (pointer to `char`) are automatically recognized by `cout`.

➔

Customer
Service

(813) 377-1121

COMPUTERS
etc!Technical
Support

(813) 378-2394

**SAVE ON AMIGA HARDWARE, SOFTWARE,
AND ACCESSORIES!****A500**

20 MB Hard Drive	495.00
40 MB Quantum HD	695.00
80 MB Quantum HD	950.00
2 MB Ram Card for HD (populated)	260.00
512K Ram Expansion	110.00

A2000

40 MB Quantum HD	625.00
80 MB Quantum HD	899.00
8 MB Ram Card w/2 MB onboard	325.00
Supra 2400 Baud Modem w/cable	129.00

Bridgeboard Speaker 19.95
(Easy to Install, adds sound to Bridgeboard!)

A500 Replacement Power Supplies

HDL-150 (150 Watt)	99.00
HDL-150DL (150 Watt, 3 a/c outlets)	129.00

*Call for More Information*1
8
0
0
6
3
4
5
5
4
6**Call for our Free 48 Page Catalog!**

Circle 113 on Reader Service card.

Now we only need to compile file box.cp to box.o (which we could place in a library for later reuse). When the box class is used in a program, we never need to look at its source code. All we need is the header file box.h and the object file box.o. Here's a short program that uses class box:

```
#include "box.h"

main()
{
    box myBox( 5, 7.8, "Empty Box" );

    myBox.fillx( 3 );
    myBox.display();
}
```

We compile this file and link it with box.o to produce an executable program that takes advantage of class box. What's striking about this program is its simplicity. First we create an object myBox of type box, the same way that you'd declare a typedef'd variable in C. The constructor box() is called automatically when object myBox is created. We call member function fillx() with a structure-like dot notation and then call display(). Finally, when box goes out of scope at the end of main(), its destructor ~box() is called automatically and box's memory is freed up. All the user must know is what actions must be done with the object - how it's done is unimportant

to the user. A program becomes a series of controlled actions performed by "asking" the object to do so. If we wish to change how the member functions work, the program will still do what it should - create an object from class box, fill it, display it, and destroy it when it's no longer in use.

Of course, C++ programs offer a lot more than the class box, or C++ would not be worth using! C++ has features that support all the advantages of object-oriented programming. However, we'd need a lot more space to demonstrate a program that includes all these features. So let's change gears and discuss how C++ is implemented on the Amiga.

C++ ON THE AMIGA

At present, the sole implementation of C++ on the Amiga is Lattice C++, version 1.0. Lattice C++ is a language translator that converts the C++ code we write into standard C code which is compiled and linked as in any C program. Although it is an implementation of an old version of C++, it does contain some Amiga-specific features not found in standard AT&T C++, such as class interfaces for the entire operating system. Lattice is also working on a true C++ compiler that bypasses intermediate C code as well as a new version of C++ that conforms to the AT&T C++ version 2.0 standard that has been implemented on MS-DOS and UNIX computers.

FEEDBACK NEEDED

This introduction to C++ barely scratches the surface of what you may do with C++. Both object-oriented programming concepts and C++ features are easy to list but require time to teach in detail. Consider this article to be a fast, simple "evangelical" description. If you want more in-depth "sermons", you must tell the editors of *Amazing Computing* that you're interested in C++! If your interest is great enough, maybe we can make this article the first of a series.

•AC•

REFERENCES

Cox B.J. (1987) *Object-oriented Programming: An Evolutionary Approach*. Addison-Wesley Publishing Company.

Lippman S.B. (1989) *C++ Primer*. Addison-Wesley Publishing Company.

Stroustrup B. (1986) *The C++ Programming Language*. Addison-Wesley Publishing Company.

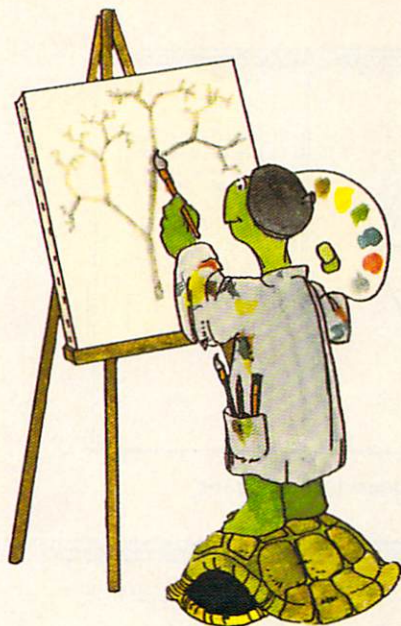
Wiener R.S. and Pinson L.J. (1988) *An Introduction To Object-oriented Programming And C++*. Addison-Wesley Publishing Company.

Lattice C++ Compiler Manual, Version 1.00, Lattice, Incorporated.

ABOUT THE AUTHOR

Dr. Scott Steinman is a Ph.D. in physiological optics who is currently conducting scientific research on the human visual system at the Smith-Kettlewell Eye Research Foundation in San Francisco. He has developed software for the Amiga since 1986.

AMIGA TURTLE GRAPHICS



Computer graphics and programming with a LOGO-like system

by Dylan McNamee

A GREAT WAY TO INTRODUCE computer graphics and programming to beginners is to teach them turtle graphics. Turtle graphics are also a great tool for both beginners and experts for playing around with fractals and other geometric constructions. Unfortunately, there are not any public domain turtle graphics packages for the Amiga. This article presents a package that implements turtle graphics on the Amiga. There are two parts to the article. The first section presents a programming project that implements a turtle graphics package. The second part is a tutorial on using the package with some examples of creating fractals with the turtle.

Turtle graphics are a simple way of drawing on a screen, using commands given to an imaginary "turtle." The turtle sits on the screen in a particular position, facing in a particular direction. Commands (such as forward 10, or right 45) given to the turtle are relative to its current position and direction. Turtle graphics are traditionally implemented as a part of the Logo language, and rarely in any other way, even though Logo and turtle graphics are not inextricably related.

PART I:

LISP TURTLE GRAPHICS—A PROJECT

This programming project implements turtles using LISP instead of Logo.

LISP is similar to Logo in many ways; LISP is actually the parent language of Logo. A turtle graphics system implemented under LISP would be part of an inherently more powerful language, useful beyond the learning years. Scheme is an implementation of LISP with some changes to make it more consistent and easier to learn. Scheme is widely used in education, often in places where Logo once was. Best of all, there is an excellent public domain version of Scheme for the Amiga, available in the Fish Disk collection disk number 149.

The project has two components—the Scheme half and the turtle half. The Scheme half is just a set of routines, written in Scheme, which set up the turtle environment, providing some interface routines for the turtle half. The turtle half is really two parts as well. There is a drawing half, called the turtle server, and a command interpreter—the turtle client, both written in C. The client sends command requests to the server, then the server interprets and executes them.

The turtle server is just a graphics window and a set of routines which draw on the window according to the commands which arrive at its message port. The commands are plain turtle commands, such as left X, forward N, penup, pendown, etc.,



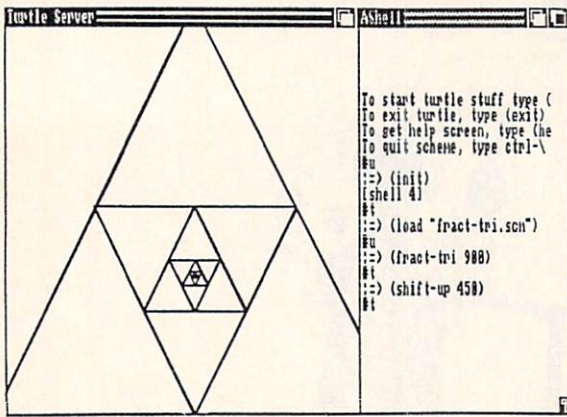


Figure One: Fractal produced by Listing Ten

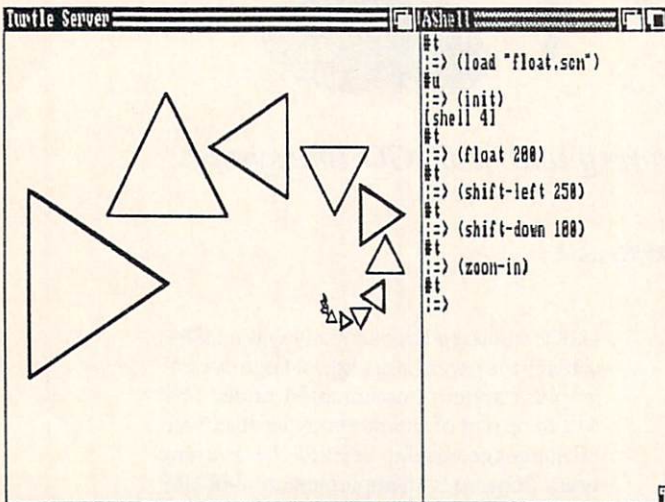


Figure Two: Demonstrating floating elements with Listing Eleven

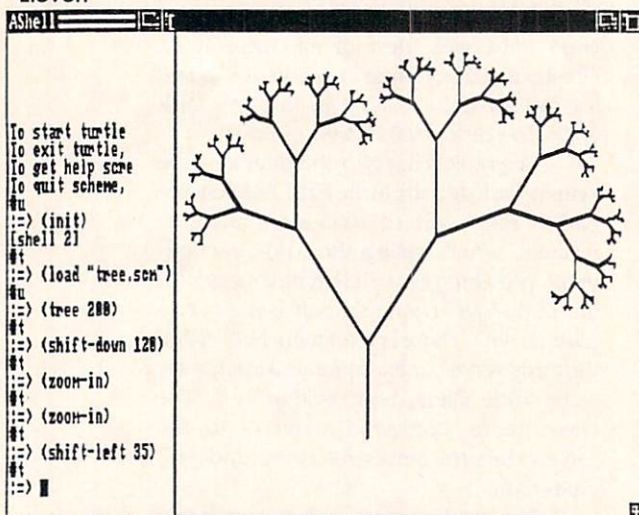


Figure Three: Tree figure produced by Listing Twelve

plus a few special display commands like zoomin, zoomout, etc.

The turtle client is a simple program which is run from the CLI and takes commands (or a file of commands) and sends them, one at a time, to the turtle server.

When a Scheme program that uses these turtles is running, a few things are really happening before a Scheme command like (forward 10) actually draws a line: in Scheme, (forward 10) calls the forward routine with the argument 10. The forward command executes an AmigaDOS system call "parse forward 10". AmigaDOS gets the "parse forward 10" command and runs the parse program with the two arguments forward and 10. Parse (the turtle client) runs and sees the command forward 10, packages up a message for the server, and sends it. The server wakes up when it receives a message, sees the forward 10 command, and using the current coordinate and angle, draws the line on the screen and updates the current coordinate.

All of this sounds like a computer version of a Rube Goldberg machine; however, there were not many other ways of doing the same thing. I could have changed Scheme itself, making the turtle client built in, but modifying the Scheme source code would be messy, and no one wants lots of special mutated Scheme's running around. Writing the commands to a PIPE: device and starting up the client like "parse < PIPE:tmp" will not work either, because AmigaDOS buffers files and the commands are not executed until a lot of them have been written, or the pipe file is closed.

One final way to do it, which was implemented as an option, is to have Scheme write its commands to a file instead of calling the turtle client each step. When you want to draw the picture, call the client (which is called parse), redirecting in the file Scheme wrote. For example, if the file was called "turtle.cmd", you would see the picture by typing "parse < turtle.cmd", and the picture would be drawn as fast as the file could be read.

THE PROGRAM

The main server module is presented in Listing One. The server keeps track of the turtle's current (x,y) coordinate with the global variables turtleX and turtleY. The server accepts commands through its message port and executes them one at a time until it receives an EXIT_TURTLE command.

Each of the actions the turtle can do is represented by a procedure in the server. Some of the commands just change a state variable; the commands right and left just modify the turtleAngle variable. Forward and Backward use the turtleAngle, turtlePenDown and the distance argument to draw the turtle's path.

The turtle server provides a few other functions as well. The refresh procedure clears the image and redraws the turtle's figure on the window. The clear screen operation resets the turtle to its original position and direction, and erases the image. The shift operations shift the whole image on the screen. Shifting is useful in conjunction with zoom in revealing parts of the image that are off the screen. As the turtle moves on the screen, each line is stored in the RefreshArray for the refresh procedure to use when it redraws the image. To zoom in or out, all that needs to be done is change the scale and call refresh.

The drawing code is in Listing Two. The main procedure in the drawing code is DrawLine. It takes the line that the turtle server told it to draw, and actually does the drawing. First it transforms the coordinates according to the current scale and horizontal and vertical offsets. Then it "clips" these coordinates to the screen. Clipping means taking a line which may extend beyond the screen's boundaries (which are stored in maxx and maxy) and drawing only the part of the line that is visible. Finally, DrawLine adds the line to the refreshArray, which is

the series of lines that get drawn when the screen is refreshed.

Listing Three contains the Intuition message passing code. It provides the procedure GetCommand to the turtle server. The corresponding Put command for the client is a direct call to Intuition's PutMsg call. Listing Four is the header file for the message code. It defines Intuition's names for the server and client, and the definition of the message object structure, ObjMsg. Intuition expects the Message structure at the top of any messages it passes, but it will also carry around data you tack onto it. In this case, the command type and its argument have been tacked on.

Listing Five is the code that handles the turtle window. It also provides the internal drawing routines MoveTo, DrawLineTo, and clearScreen used by the draw routines. If we wanted the turtle graphics to be drawn to a custom screen rather than a workbench window, this is the file to modify; we would change InitGraphics to open a screen instead of a window, and modify the drawing routines to use the screen's RastPort instead of the window's RastPort. Listing Six is a collection of header files for the window and drawing functions.

Listing Seven defines each of the turtle commands, and the Command structure the server gets from GetCommand. These definitions are shared between the server and the client.

Listing Eight is the client program, parse. This program has two modes of input. If there are command line arguments, they are taken to be the command, otherwise, it accepts a stream of commands from standard input (the keyboard or a file redirected in from the CLI, as in "parse < command.file").

A case statement parses the current command, and packages up a command message for the server. If the command takes an argument, it reads it into the argument field of the message structure. After the message is all packaged up, it sends it to the server with Intuition's PutMsg function.

The client program is very simple because all of the code that actually deals with the turtle, the window, and drawing is all in the server. It would be fairly easy to write other turtle clients. A control panel with buttons for the turtle commands and sliders for the zooming and shifting functions would be fairly easy to write and integrate with the server.

All that is left is the Scheme support file for the turtle graphics. This is Listing Nine. I put this file in the S: directory and called it scheme-init.scm, so it gets loaded each time I run Scheme. To start using the turtles, just type (turtle-init) and a new turtle server shows up, and now you are ready to start programming the turtles in Scheme.

PART II. USING LISP TURTLES—AN INTRODUCTION TO RECURSION USING FRACTALS

One of the best ways to learn about the technique of recursion is to try it, watch the program work, and then figure out what is going on to make it work. Learning recursion by watching the turtles execute a recursive routine is much easier than studying a trace of a non-graphical recursive program.

Turtle graphics provide a natural way to express recursive drawings. The most famous recursive drawings are fractals like the Mandelbrot Set, which have been presented in *Amazing Computing* in the past. These fractals rely on

some amazingly chaotic properties of simple functions. One drawback of these fractals is that they produce a fixed design; you cannot design your own shapes. With turtles, designing your own recursive fractals is amazingly easy.

My sister's 7th grade computer class had been learning Logo and turtle graphics for almost a year when I visited. After seeing all their iterative turtle graphic images, I asked if they had learned recursion. The instructor said no; it was hard enough to teach to college students. I went home and designed a simple fractal in Logo using recursion, and explained it to my sister. She said "it's so simple!" The next day she presented the program, and taught recursion to the class. The fractal I designed that night is in Listing Ten, which produces Figure One.

Making fractals using turtles is really easy. First, you draw a shape, using the size passed in as an argument. Then have the routine call itself with a smaller size one or more times somewhere where you want it. Finally you return the turtle to the same place it started, and point it in the same direction. Usually, as I have done, you should check to see if the size is above a certain threshold, before drawing, preventing the presence of an infinite loop.

Fractals do not have to be connected shapes, they can have floating elements, like Listing Eleven, which made the fractal in Figure Two. There is no reason the fractals have to be geometric shapes. Listing Twelve is the program that drew the tree in Figure Three.

EXTRA FEATURES OF AMIGA TURTLES

If the drawing in the turtle window is too small or too large, use the zoomin or zoomout commands (you can abbreviate them to zi and zo). If what you want to look at is off the bottom of the screen, you can shift it up with su. Shifting down, left and right are also implemented, with sd, sl and sr respectively. For reference, these functions are all defined in the scheme-init.scm file in Listing Nine.

I hope this article inspires you to create some turtle fractals of your own. Turtle graphics for the Amiga should help a new generation of Amiga programmers learn that LISP and recursion are easy and fun.

FURTHER REFERENCES

There are many excellent books on programming with Turtles. Most of them, even though they are based on Logo, will apply directly to Amiga Turtles as well. An excellent book on learning Scheme is called the Structure and Interpretation of Computer Programs by Abelson, Sussman and Sussman, from MIT Press.

ACKNOWLEDGEMENTS

I'd like to thank Ted Griggs for helping me with Intuition messages. Thanks also to everyone who has submitted programs to the Amiga public domain.

Listing One

```
/*
  Listing 1: turtle.c
  main module for turtle graphics Server
*/
#include <math.h>
#include "cmds.h"
/* pi / 180. For converting degrees to radians */
#define PID180 0.0174532925199

/* direction (in degrees) turtle is facing */
float turtleAngle;
/* turtle position */
float turtleX, turtleY;
/* boolean. True == pen is down (drawing) */
short turtlePenDown;

/* from draw.c */
extern float scale;
```



```

extern int xoff, yoff;
extern int resetScreen(), refPtr;

void main(ac,av)
int ac;
char *av[];
{
    Command command;

    turtleAngle = 180.0;
    turtleX = 0.0;
    turtleY = 0.0;
    penDown();
    InitGraphics();
    InitMessagePort();

    /* loop, process commands until the exit command */
    do {
        /* process command */
        GetCommand(&command);
        switch(command.type) {
            case FORWARD:
                forward(command.argument);
                break;
            case BACKWARD:
                backward(command.argument);
                break;
            case LEFT:
                left(command.argument);
                break;
            case RIGHT:
                right(command.argument);
                break;
            case PENUP:
                penUp();
                break;
            case PENDOWN:
                penDown();
                break;
            case ZOOMIN:
                scale = scale*1.5;
                refresh();
                break;
            case ZOOMOUT:
                scale = scale/1.5;
                refresh();
                break;
            case REFRESH:
                refresh();
                break;
            case CLEARSCREEN:
                turtleAngle = 180.0;
                turtleX = 0.0;
                turtleY = 0.0;
                resetScreen();
                break;
            case HSHIFT:
                xoff = xoff + command.argument;
                refresh();
                break;
            case VSHIFT:
                yoff = yoff + command.argument;
                refresh();
                break;
            default:
                printf("Turtle Server Exiting\n");
                command.type = EXIT_TURTLE;
                break;
        }
    } while (command.type != EXIT_TURTLE);
    KillMessagePort();
    UnInit();
}

/*
    Turtle command implementation follows:
*/

penUp()
{
    turtlePenDown = 0;
}

penDown()
{

```

```

    turtlePenDown = 1;
}

right(angle)
int angle;
{
    turtleAngle = turtleAngle - angle;
    /* negative angles get converted equivalent positive */
    while(turtleAngle < 0) {
        turtleAngle += 360.0;
    }
}

left(angle)
int angle;
{
    turtleAngle = turtleAngle + angle;
    /* if angle exceeds 360, wrap around to 0 degrees */
    while (turtleAngle > 360) {
        turtleAngle -= 360.0;
    }
}

forward(dist)
int dist;
{
    float angle;
    float dx, dy;
    int x,y,nx,ny;

    angle = turtleAngle * PID180;

    /* convert polar coordinates to cartesian */
    dx = (dist * sin(angle));
    dy = (dist * cos(angle));

    /* only draw if the pen is down */
    if (turtlePenDown) {
        x = (int)(turtleX + 0.5);
        y = (int)(turtleY + 0.5);
        nx = (int)(turtleX + dx + 0.5);
        ny = (int)(turtleY + dy + 0.5);
        DrawLine(x,y,nx,ny);
    }

    turtleX += dx;
    turtleY += dy;
}

/*
    backing up is implemented as turning around, moving
    forward, and turning around again to save code.
*/
backward(dist)
int dist;
{
    right(180);
    forward(dist);
    left(180);
}


```

Listing Two

```

/* Listing 2: draw.c
    graphics drawing, clipping, scaling and refreshing
    routines.
*/

/* Out of bounds defines. Can be logically OR'd together */
#define TOP 8
#define BOTTOM 4
#define RIGHT 2
#define LEFT 1

/* number of lines to save in refresh array */
#define MAXREFRESH 2000

/* aspect ratio of workbench screen. This way, squares are
square and circles are not ellipses */
#define ASPECT 2

/* screen bounds. These are variables, and get updated when
the window is resized */
int minx = 0;
int miny = 0;
int maxx = 640;
int maxy = 200;

```



```

extern int clearScreen(); /* from window.c */

typedef struct LineElt {
    long int x1,y1,x2,y2;
} LineElt;

LineElt refreshArray[MAXREFRESH]; /* for redrawing screen */

int refPtr = 0; /* index into refresh array */
float scale = 0.25;

/* center and offsets for window */
int xc = 320;
int yc = 100;
int xoff = 0;
int yoff = 0;

/* draws a line, and adds to the refresh array */
DrawLine(x1,y1,x2,y2)
int x1,y1,x2,y2;
{
    int nx1,ny1,nx2,ny2;

    nx1 = (x1 + xoff) * scale * ASPECT + xc;
    ny1 = (y1 + yoff) * scale + yc;
    nx2 = (x2 + xoff) * scale * ASPECT + xc;
    ny2 = (y2 + yoff) * scale + yc;

    clip(nx1,ny1,nx2,ny2);
    addRefresh(x1,y1,x2,y2);
}

/* resetScreen resets scale, offset and the refresh array,
and clears the screen */
resetScreen()
{
    refPtr = 0;
    scale = 0.25;
    xoff = 0;
    yoff = 0;
    clearScreen();
}

/* clip(x1,y1,x2,y2) - internal routine
clip takes a line segment and draws the portion which lies
within the window */

clip(x1,y1,x2,y2)
int x1,y1,x2,y2;
{
    int c1,c2,c,x,y;

    c1 = code(x1,y1);
    c2 = code(x2,y2);

    while((c1 != 0) || (c2 != 0)) {
        if ((c1 & c2) != 0) return; /* don't even draw line */
        c = c1;
        if (c == 0) c = c2;
        if (c & LEFT) { /* out of bounds to the left */
            y = y1 + (y2 - y1)*(minx - x1)/(x2-x1);
            x = minx;
        }
        if (c & RIGHT) { /* out of bounds to the right */
            y = y1 + (y2 - y1)*(maxx - x1)/(x2-x1);
            x = maxx;
        }
        if (c & BOTTOM) { /* out of bounds to the bottom */
            x = x1 + (x2 - x1)*(miny - y1)/(y2 - y1);
            y = miny;
        }
        if (c & TOP) { /* out of bounds to the top */
            x = x1 + (x2 - x1)*(maxy - y1)/(y2 - y1);
            y = maxy;
        }
        if (c == c1) { /* (x1,y1) was out of bounds */
            x1 = x;
            y1 = y;
            c1 = code(x,y); /* recheck bounds */
        }
        else { /* (x2,y2) was out of bounds */
            x2 = x;
            y2 = y;
            c2 = code(x,y); /* recheck bounds */
        }
    }
    MoveTo(x1,y1);

```

We take a **byte** out of the price!



ONE BYTE

P.O. Box 455
Quaker Hill, CT 06375
(800) 441-BYTE, in CT (203) 443-4623

YOUR ONE-STOP AMIGA STORE

**Authorized dealer for
Commodore-Amiga Computers,
Great Valley Products (GVP),
Authorized Commodore-Amiga Service and Repair.
Authorized Amiga Graphics Dealer.**

Circle 135 on Reader Service card.

AMIGA IS A REGISTERED TRADEMARK OF COMMODORE-AMIGA, INC.

```

DrawLineTo(x2,y2);
}

/* code(x,y) - internal routine
returns which limits (if any) a point is out of bounds of
*/
int code(x,y)
int x,y;
{
    int c;
    c = 0;
    if (x < minx)
        c = LEFT;
    else if (x > maxx)
        c = RIGHT;
    if (y < miny)
        c = c + BOTTOM;
    else if (y > maxy)
        c = c + TOP;
    return(c);
}

/* clears and redraws the screen's lines, according to
the current scale and offset */
void refresh()
{
    int i,x1,y1,x2,y2;

    clearScreen();
    for(i = 0; i < refPtr; i++){
        x1 = (int)(refreshArray[i].x1 + xoff) * scale * ASPECT + xc;
        x2 = (int)(refreshArray[i].x2 + xoff) * scale * ASPECT + xc;
        y1 = (int)(refreshArray[i].y1 + yoff) * scale + yc;
        y2 = (int)(refreshArray[i].y2 + yoff) * scale + yc;
        clip(x1,y1,x2,y2); /* note: don't want to add to refresh
array */
    }
}

```


BRIDGEBOARD USERS!

Don't waste money, slots, or desk space buying extra IBM-compatible or Amiga floppy drives! The **Bridge Drive Commander+** gives you direct access to all your internal and external Amiga drives from the Bridgeboard, and direct access to IBM type 360K and 720K drives from AmigaDOS. **Bridge Drive Commander+** is totally transparent and automatic. Put an IBM type disk in any drive and use it just like on any IBM compatible! Put in an Amiga disk and return to Amiga use! Just that simple, just that fast! One drive can use Amiga disks at the same time another is using IBM-compatible disks. Disks are completely usable by other Amiga and IBM-compatible computers. All hardware, no software drivers to load, no precious memory or expansion slots used up. Plugs onto motherboard at internal drive connector. (No soldering or wiring changes.) Compatible with all Bridgeboards (8088, 80286), SideCar, all accelerator boards (any 680x0), hard disks and other hardware and software.

Bridge Drive Commander+\$ 97.50

MJ SYSTEMS

Dept 10A, 1222 Brookwood Road, Madison, WI 53711

1-800-448-4564

(24 hours MasterCard/VISA)

Product names are trademarks of their respective companies.

Circle 149 on Reader Service card.

```
}
}

/* addRefresh(x1,y1,x2,y2) - internal routine
   adds a line to the refresh array, clears the array on
   overflow */
addRefresh(x1,y1,x2,y2)
int x1,y1,x2,y2;
{
    refreshArray[refPtr].x1 = x1;
    refreshArray[refPtr].x2 = x2;
    refreshArray[refPtr].y1 = y1;
    refreshArray[refPtr].y2 = y2;
    if (refPtr > 2000) refPtr = 0; /* start at zero again */
}
```

Listing Three

```
/* Listing 3: msgs.c
   Intuition message handling code */
#include "msgs.h"
#include "cmds.h"
#include "window.h"

struct MsgPort *sport;
ULONG ipcbit, intuibit;

extern struct Window *wdw;
extern struct Screen *scr;
extern int minx,maxx,miny,maxy,xc,yc;
extern float scale;

InitMessagePort()
{
    if((sport=CreatePort(SERVEPORT,0))==NULL) {
        printf("Cannot Create Server!\n");
        exit(1);
    }

    ipcbit=(1L<<sport->mp_SigBit);
    intuibit=(1L<<wdw->UserPort->mp_SigBit);
```

```
}

KillMessagePort()
{
    RemPort(sport);
    FreeMem(sport,sizeof(struct MsgPort));
}

GetCommand(message)
Command *message;
{
    ObjMsg *msg;
    struct IntuiMessage *intuiMsg;
    ULONG bits;

    /* intuition returns status bits */
    bits=Wait(ipcbit | SIGBREAKF_CTRL_C | intuibit);

    if(bits & intuibit) {
        /* we got a message from intuition */
        while(intuiMsg = GetMsg(wdw->UserPort)){
            if (intuiMsg->Class == NEWSIZE) { /* User has resized
            window */
                /* new scale is determined by window's smallest
            dimension x or y */
                if(wdw->Width < wdw->Height)
                    scale = (scale * wdw->Width) / maxx;
                else
                    scale = (scale * wdw->Height) / maxy;
                maxx = wdw->Width; /* update our maxx and maxy */
                maxy = wdw->Height;
                xc = maxx / 2; /* update center coordinate */
                yc = maxy / 2;
                message->type = REFRESH; /* ask server to refresh
            window */
            }
            ReplyMsg(intuiMsg);
        }
        return(1);
    }

    /* handle a user command */
    if((msg=(ObjMsg *)GetMsg(sport))==NULL) return(0);
    message->type = msg->type;
    message->argument = msg->argument;
    ReplyMsg(msg);
    return(1);
}
```

Listing Four

```
/* Listing 4: msgs.h
   defines turtle message, which starts with
   an EXEC message, then contains the stuff I need
   for passing turtle messages */
#include <libraries/dos.h>
#include <exec/ports.h>

/*
 * Name of object to object communication server
 */
#define SERVEPORT "TurtleServer"
#define CLIENTPORT "TurtleClient"

/*
 * This structure is passed by PutMsg to the server.
 */
typedef struct ObjMsg {
    struct Message Msg;
    int type;
    int argument;
} ObjMsg;
```

Listing Five

```
/* Listing 5: window.c
   sets up a graphics window, provides DrawLine,
   MoveTo and clearScreen
 */
#include "window.h"
extern struct IntuitionBase *IntuitionBase;
struct GfxBase *GfxBase;

#define INTUITION_REV 29
#define GRAPHICS_REV 29

#define STRLEN 30 /* String length */
#define BP 4 /* Number of bit-planes */
```



```

#define SWIDTH 320      /* Screen width */
#define SHEIGHT 200    /* Screen height */

struct TextAttr MyFont = {
    "topaz.font",
    TOPAZ_EIGHTY,
    FS_NORMAL,
    FFF_ROMFONT,
};

struct NewWindow NewWindow = {
    0,0, /* initial position */
    640,200, /* bitmap size */
    3, 1, /* pen numbers */
    NEWSIZE, /* intuition message flags */
    SMART_REFRESH | ACTIVATE |
        WINDOWDEPTH | WINDOWDRAG | WINDOWSIZING,
    NULL,
    NULL,
    "Turtle Server",
    NULL,
    NULL,
    48,25, /* minimum size */
    640,400, /* max size */
    WBENCHSCREEN /* screen type */
};

struct Window *wdw;

InitGraphics()
{
    IntuitionBase = (struct IntuitionBase*)

    OpenLibrary("intuition.library", INTUITION_REV);
    GfxBase = (struct
GfxBase*)OpenLibrary("graphics.library", GRAPHICS_REV);
    if((wdw = (struct
Window*)OpenWindow(&NewWindow))==NULL) exit(1);
    SetColor(1);
}

SetColor(color)
int color;
{
    SetAPen(wdw->RPort, color);
}

UnInit()
{
    CloseWindow(wdw);
}

/* Graphics drawing functions: */
DrawLineTo(x,y)
int x,y;
{
    Draw(wdw->RPort, x,y);
}

MoveTo(x,y)
int x,y;
{
    Move(wdw->RPort, x,y);
}

clearScreen()
{
    Move(wdw->RPort, 0,0);
    ClearScreen(wdw->RPort);
    RefreshWindowFrame(wdw);
}

```

Listing Six

```

/* Listing 6: window.h
Header files
*/
#include <stdio.h>
#include <libraries/dos.h>
#include <exec/ports.h>
#include <exec/types.h>
#include <exec/exec.h>
#include <intuition/intuition.h>
#include <intuition/screens.h>
#include <intuition/intuitionbase.h>
#include <hardware/dmabits.h>
#include <hardware/custom.h>
#include <hardware/blit.h>

```

Come see whats Hot for the *AMIGA* at The Memory Location



396 Washington Street
Wellesley, MA 02181
(617) 237-6846

AMIGA Experts! Nothing but the best.
Satisfaction guaranteed.

**Come in and try-out all the latest software,
hardware, and accessories.**

Your full service *AMIGA* dealer.

Store hours: Mon.-Thur. 10-6 Friday 10-8 Saturday 9-5.
Full Commodore authorized repair service onsite.
Low flat rate plus parts.

Circle 107 on Reader Service card.

```

#include <graphics/gfx.h>
#include <graphics/gfxmacros.h>
#include <graphics/rastport.h>
#include <graphics/view.h>
#include <proto/exec.h>
#include <proto/intuition.h>
#include <fcntl.h>
#include <math.h>

```

Listing Seven

```

/* Listing 7: cmds.h
Commands definition file.
Defines turtle commands for server and client.
Defines Command structure.
*/

```

```

#define FORWARD 1
#define BACKWARD 2
#define RIGHT 3
#define LEFT 4
#define PENUP 5
#define PENDOWN 6
#define ZOOMIN 7
#define ZOOMOUT 8
#define HSHIFT 9
#define VSHIFT 10
#define REFRESH 11
#define CLEARSCREEN 12
#define EXIT_TURTLE 127

```

```

typedef struct Command {
    int type; /* right, left, forward, etc */
    int argument; /* argument for command, if applicable */
} Command;

```

Listing Eight

```

/* Listing 8: parse.c
Turtle client program.

```


The Power Is Amazing.

Only
Amazing/AMIGA
COMPUTING
Your Original AMIGA® Monthly Resource

gives you
the power to do more
with your Amiga!

In recent issues,
AC has given you the power
to turn your A1000 into a
ROM-based machine.

To upgrade your A500 to 1 megabyte.

To transform your Amiga into a
biofeedback/lie detector device.

To use your Amiga as
the command center for a
remote-controlled home of the future.

A future that is
in your hands now,
thanks to **Amazing Computing!**

There's plenty of power in our product reviews and tutorials, too, and **AC** features more informative monthly columns than *any other Amiga publication!*

AC previews important new products that are *really* new, and takes you to all the major shows with complete, timely coverage!

We also put *the entire Amiga marketplace* at your fingertips in the pages of

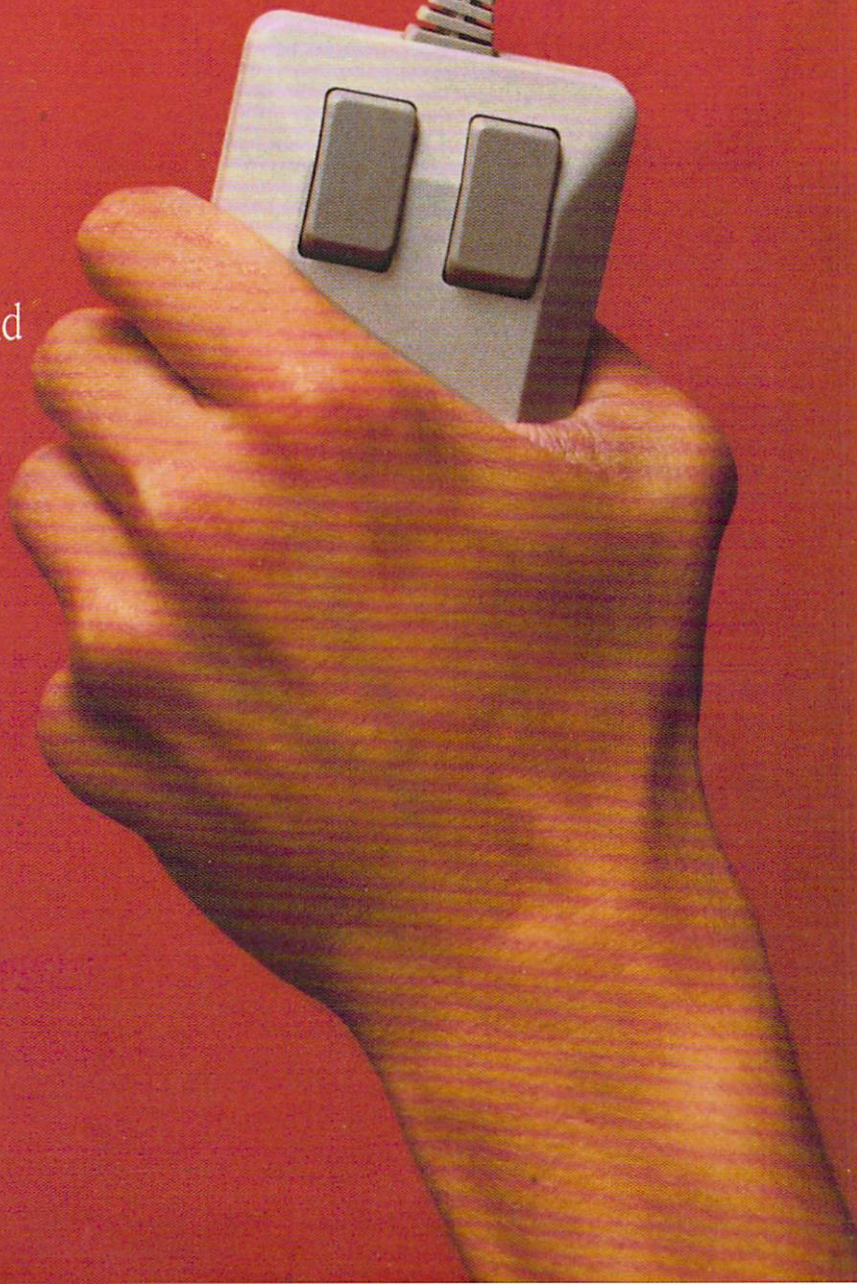
AC's GUIDE/AMIGA

It's the only publication in the world that lists *every product* currently available for the Amiga!

Put **AC POWER** in your hands and save over 48% off the newsstand price of the next 12 monthly issues of **AC**, **PLUS AC's Spring Guide!**

Use one of the cards provided in this powerful issue to become a subscriber!
Or, just call 1-800-345-3360 (credit card orders only)!

Subscribe Today!



The Krueger Company
can provide a wide
variety of **MEMORY,**
MICROPROCESSOR,
and **SUPPORT CHIPS.**

ALL **KRUEGER** PARTS
carry an unconditional 30 day
guarantee. The minimum
purchase is \$50.00.

Call Myron Lieberman at
(800)245-2235 or FAX
(602)820-1707. Mention
AMAZING COMPUTING.



ANOTHER TIP
FROM DR. CHIP



8 MHZ 68010	\$ 1.00
12 MHZ 68020	\$30.00
16 MHZ 68030	\$40.00
5 MHZ 8088	\$ 1.00
8 MHZ 8086	\$ 1.50
8 MHZ 80286	\$10.00
DP8464BV3 PLCC	\$ 1.00

ASK ABOUT Z80s AT GREAT PRICES

The Krueger Company (800)245-2235.

Circle 118 on Reader Service card.

Parses input stream, and sends messages
to turtle server.

```

*/
#include <stdio.h>
#include <string.h>
#include "msgs.h"
#include "cmds.h"

ULONG ipcbit;
short command_line;

main(argc,argv)
int argc;
char **argv;
{
    ObjMsg *msg;
    int args;
    struct MsgPort *sport, *cport;
    char token[50];

    /* initialize port stuff */
    msg=(ObjMsg *)AllocMem(sizeof(ObjMsg),0);
    if((cport=CreatePort(CLIENTPORT,0)) == NULL){
        printf("Cannot Create Client Port!\n");
        exit(1);
    }
    if((sport=FindPort(SERVEPORT))==NULL) {
        printf("Cannot Locate Server Port!\n");
        exit(1);
    }

    /* bits to wait on for Intuition messages */
    ipcbit = (1L << cport->mp_SigBit);

    token[0] = 42;
    command_line = NULL;

    if(argc>1) /* we've got a command line command */

```

```

command_line = 1;
}

/* loop until gettoken returns with nothing */
while (token[0] != NULL) {
    msg->Msg.mn_Length=sizeof(ObjMsg);
    msg->Msg.mn_ReplyPort = cport;

    args = 1; /* default number of arguments */

    gettoken(token,argv);

    switch(token[0]){
        case 'f':
            msg->type = FORWARD;
            break;
        case 'b':
            msg->type = BACKWARD;
            break;
        case 'r':
            msg->type = RIGHT;
            break;
        case 'l':
            msg->type = LEFT;
            break;
        case 'z':
            args = 0;
            msg->type = ZOOMIN;
            break;
        case 'x':
            args = 0;
            msg->type = ZOOMOUT;
            break;
        case 'c':
            args = 0;
            msg->type = CLEARSCREEN;
            break;
        case '6':
            msg->type = HSHIFT;
            break;
        case '4':
            msg->type = HSHIFT;
            args = 0;
            gettoken(token,argv);
            msg->argument = -1 * atoi(token);
            break;
        case '2':
            msg->type = VSHIFT;
            break;
        case '8':
            msg->type = VSHIFT;
            args = 0;
            gettoken(token,argv);
            msg->argument = -1 * atoi(token);
            break;
        case 'p':
            args = 0;
            msg->argument = -1;
            if(token[3] == 'u')
                msg->type = PENUP;
            else
                msg->type = PENDOWN;
            break;
        case 'e':
        case 'q':
            msg->type = EXIT_TURTLE;
            PutMsg(sport,msg);
            exit(0);
        default:
            continue;
    }

    if (args == 1) { /* if there's an argument to get, get it.
*/
        gettoken(token,argv);
        if (token[0] == NULL) {
            printf("Unexpected End Of File! (I was
expecting a
number)\n");
            break;
        }
        msg->argument = atoi(token);
    }

    PutMsg(sport,msg); /* send the command to the server port.
*/
    Wait(ipcbit); /* wait for the next message */

```



```

    if (command_line) break;
}

gettoken(tok,argv)
char *tok,**argv;
{
    int i;
    char c;

    /* get the tokens from argv if it's a command line call */
    if(command_line){
        strcpy(tok,argv[command_line]);
        command_line++;
        return;
    }

    c = getchar();
    if (c == EOF){
        tok[0] = NULL;
        return;
    }

    /* consume any non-alpha chars */
    while (!(isalnum(c))) {
        c = getchar();
        if (c == EOF){
            tok[0] = NULL;
            return;
        }
    }

    /* now fill the token */
    i = 0;
    while (isalnum(c)) {
        tok[i++] = c;
        c = getchar();
        if (c == EOF) {
            tok[0] = NULL;
            return;
        }
    }
    tok[i++] = 0;
}

```

Listing Nine

```

;; Listing 9: scheme-init.scm
;; turtle initialization and support routines
(define file-io nil)
(define turtle-port nil)

(display "To start turtle stuff type (init)" (newline))
(display "To exit turtle, type (exit)" (newline))
(display "To get help screen, type (help)" (newline))
(display "To quit scheme, type ctrl-\ (or (abort-system))" (newline))

(call-system "copy parse ram:parse")
;; use resident instead of copy, if available
(define (init)
    (call-system "run turtle")
)

(define (tell-command string)
    (if file-io
        (tell-file string turtle-port)
        (call-system (string-append "ram:parse " string))
    )
)

(define (tell-file string port)
    (write string port)
    (newline port)
)

(define (file-off)
    (if file-io
        (close-output-port turtle-port)
        ()
    )
    (set! file-io nil)
)

(define (file-on name)
    (if (not file-io)
        (begin
            (set! file-io #t)
            (set! turtle-port (open-output-file name))
        )
    )
)

```

NEW- F-BASIC 2.0 -NEW

The reviewers have labeled **F-BASIC**:

The **FASTEST** Growing
FASTEST Performing
AMIGA Language

Beats **C** And Other
Basics!

Available Soon... **VERSION 3.0 F-BASIC** NEW-Integrated Editor
NEW-020/881 Support
NEW-Even Faster
SAVE! Buy 2.0 Now And Upgrade To 3.0 Upon Availability For Less Money!

*A Beginner
Can Immediately Use
F-BASIC

*An Expert Can **NEVER** Outgrow **F-BASIC**

F-BASIC 2.0 With User's Manual & Sample Programs Disk—Only \$89.95
F-BASIC 2.0 With Complete Source Level Debugger—Only \$149.95

F-BASIC is Available Only From:

DELPHI NOETIC SYSTEMS, INC.

Post Office Box 7722
Rapid City, SD 57709-7722

Send Check or Money Order, or Write For Info
Credit Card or C.O.D., Call (605) 348-0791

F-BASIC is a registered trademark of DNS, Inc.
AMIGA is a registered trademark of Commodore/AMIGA, Inc.

Circle 142 on Reader Service card.

```

        (display "output file already open")
    )
)

(define (penup)
    (tell-command "penu")
)

(define (pendown)
    (tell-command "pend")
)

(define (right angle)
    (define command (string-append "r " (number->string angle)
        '(int)))
    (tell-command command)
)

(define (left angle)
    (define command (string-append "l " (number->string angle)
        '(int)))
    (tell-command command)
)

(define (forward dist)
    (define command (string-append "f " (number->string dist)
        '(int)))
    (tell-command command)
)

(define (back dist)
    (define command (string-append "b " (number->string dist)
        '(int)))
    (tell-command command)
)

(define (zoom-in)
    (tell-command "z")
)

```



```

(define (zoom-out)
  (tell-command "x")
)

(define (shift-up dist)
  (define command (string-append "8 " (number->string dist
' (int))))
  (tell-command command)
)

(define (shift-down dist)
  (define command (string-append "2 " (number->string dist
' (int))))
  (tell-command command)
)

(define (shift-left dist)
  (define command (string-append "4 " (number->string dist
' (int))))
  (tell-command command)
)

(define (shift-right dist)
  (define command (string-append "6 " (number->string dist
' (int))))
  (tell-command command)
)

;; exit and new call parse, because they
;; shouldn't be called in batch mode.
(define (exit)
  (call-system "parse quit")
)

(define (new)
  (call-system "parse c")
)

(define (help)
  (display "Turtle Graphics help:") (newline)
  (display "available commands") (newline)
  (display
    "(init)                -start up turtle
graphics")
    (newline)
  (display
    "(new)                -clears screen") (newline)
  (display
    "(forward x) or fd    -moves turtle forward x units"
    (newline)
  (display
    "(backward x) or bk") (newline)
  (display
    "(right x) or rt      -turns turtle clockwise x
degrees")
    (newline)
  (display
    "(left x) or lt") (newline)
  (display
    "(zoomin) or zi       -zooms in on center of screen"
    (newline)
  (display
    "(zoomout) or zo      -zoom out") (newline)
  (display
    "(shift-up x) or su   -moves image up x units"
    (newline)
  (display
    "(shift-down x) or sd -moves image down x units"
    (newline)
  (display
    "(shift-left x) or sl -moves image left x units"
    (newline)
  (display
    "(shift-right x) or sr -moves image right x units"
    (newline)
  (display "") (newline)
)

```

Listing Ten

isting 10:

```

(define (fract-tri size)
  (if (> size 2)
    (begin
      (right 150)
      (forward size) ;; right side of the triangle
      (right 120)
      (forward (/ size 2)) ;; half of the bottom side
      (left 90)
      (fract-tri (/ size 2)) ;; smaller upside down triangle
    )
    here
  )
  (right 90)
  (forward (/ size 2)) ;; other half of the bottom
  (right 120)
  (forward size) ;; left side of the triangle
  (left 30) ;; make sure we're pointing straight up
  again
)

() ;; if size <= 5 do nothing
)

```

Listing Eleven

```

Listing Eleven
(define (float size)
  (if (> size 3) ;; if size is big enough, draw the triangle
    (begin
      (triangle size) ;; draw a full size
      triangle
      (right 30) ;; go off to the right
      (forward size)
      (float (/ (* size 3) 4)) ;; smaller, recursive call
      (back size)
      (left 30)
    )
    () ;; if the size is too small, do nothing
  )
)

(define (triangle size)
  (pendown)
  (forward size)
  (right 120)
  (forward size)
  (right 120)
  (forward size)
  (right 120)
  (penup)
)

```

Listing Twelve

```

;; Listing 12: tree.scm
;; fractal tree
(define (tree size)
  (if (> size 3)
    (begin
      (forward (/ size 3)) ;; draw the trunk
      (right 36)
      (forward (/ size 4)) ;; the right branch
      (tree (/ (* size 2) 3)) ;; limbs to the right
      (back (/ size 4)) ;; back up right branch
      (left 72)
      (forward (/ size 3)) ;; left branch is bigger
      (tree (/ size 2)) ;; limbs to the left
      (back (/ size 3)) ;; back to the trunk
      (right 36) ;; straighten out
      (back (/ size 3)) ;; back to the start point
    )
    ()
  )
)

```


APL + THE AMIGA

Primitive Functions and their Execution

by Henry T. Lippert, EdD

APL IS A LANGUAGE THAT YOU CAN USE. SO MANY OF THE OTHER COMPUTER LANGUAGES REQUIRE that one must know the entire language structure in addition to linkers, loaders, the CLI, DOS, and so forth, in order to do anything useful. APL is appropriate for the beginner, a child, an adult—anybody! It is also a language that can be used to write an airline reservation system where hundreds or perhaps thousands of terminals access a common data base. While it is an all-purpose language, it may not always be the best choice for all purposes. It is an interpretive language. That means that each statement is stored in the program just as you wrote it and is not converted to machine language until it is ready for execution. The code is said to be “interpreted.” For extremely fast applications, such as the airline reservations problem, APL would probably be a poor choice. It would probably prove to be slow. The general APL answer might be that although the execution speed is not as fast as compiled code, the fact that it is array oriented and can do vast matrix operations with very little code to be interpreted, may make it the fastest game around. The point is that APL is a complete language, with all the bells and whistles that anybody could ever want. It will probably always remain a special language, and it may not be the all-purpose language for all uses. The language and its capabilities, however, will not prove to be a limiting factor for any programmer, no matter what the application. Let us return to the business of how one writes instructions to drive a computer.

An APL program is called a “function.” In the last article we wrote a function that computed the arithmetic mean of a group of numbers. The function listing was:

```
∇ R ← Average X
[1] R ← (+/X) ÷ ρ X ∇
```

and used one of the six ways that an APL function can be written. The one used required only a right argument and produced an explicit result. The following is a “call” to the function with a set of numbers as the right argument:

```
AVERAGE 1 6 7 4 3 5
4.33333333
```

where APL returned the result on the next line.

The symbol used for comments or remarks is the lamp (⌘), and it is used for “illumination” in the form of a comment or a documented action. No attempt is made to execute that which follows the lamp symbol. While APL has often been criticized about lack of or difficulties with documentation, it

is more often the lack of use of the documentation tools that are provided rather than a deficiency of the language. The lamp will be used on lines of future functions for the purpose of documenting the purpose and action of the APL statement.

Let's review. The first use of left arrow was to specify variables:

```
TAX ← .0725 ⌘ SPECIFY A SCALAR, THE TAX RATE
PRICES ← .37 6.34 27.98 48.00 ⌘ A VECTOR OF PRICES
```

The second use of the left arrow was in the definition of a program to specify an explicit result, as in the header of the program AVERAGE above.

The slash “/” was called “reduction” and was used to extend a primitive mathematics function to each element of the array to its right. In executing the operation, a reduction in the rank of the variable took place, i.e., a vector became a scalar:

```
+ / SALARIES ⌘ ADD UP ALL THE SALARIES
```

The Greek letter for small r was “rho” (ρ) and was used to find the shape of its right argument:

→

indicating that there are four prices in the list stored at PRICES. Let's try it (shape) on the tax rate:

pTAX

and nothing happened. Apparently APL did nothing. Actually, APL returned an empty line. Hmmm. Think of it this way. The variable TAX contains a single number, a scalar. It is a point on the number line (at .0725) and, as a point, has no shape or dimension. APL, therefore, when asked for the shape the scalar, returned an empty vector, i.e. TAX is a single quantity with no dimension.

Many of the APL primitive functions have dual definitions depending upon whether they are used with one right argument or with two arguments, one on each side. Monadic rho, for example, returns the shape of its right argument. What happens if it is used diadically? Let's try.

```
⊞ ← matrix ← 3 4 p 1000
1000 1000 1000 1000
1000 1000 1000 1000
1000 1000 1000 1000
```

It "reshapes" its right argument according to the left argument. In this case we asked for a "three by four reshape of the number 1000." This is the way that a matrix is created. APL will handle arrays up to eight dimensions! It is, however, rare to use more than three or four dimensions.

OK, let's look at some new primitive functions that we can now use in APL.

The comma (,) is called "ravel" which no doubt came from the word "unravel." In APL, it does just that; it (un)ravels its right argument.

, Matrix

```
1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000
```

Previously, when we got the empty vector as the "dimension" of a scalar TAX, we knew that the variable was a scalar. Let's ask for the dimension of the same scalar that has been raveled

, TAX

1

OK, the raveled TAX now has a dimension of one, but since it is a single number (0.725), it now becomes a one element vector. Note that the request for the dimension of TAX:

pTAX

reveals that it is still a scalar. What happened? Nothing, which is correct. Note that we asked (two statements above) for the dimension of the **raveled** variable TAX. Contrast that with the following statement:

TAX ← , TAX

and what happens when we ask for the dimension of TAX.

pTAX

1

It was changed into a one element vector. This time we said to APL, "TAX is **specified** by the raveled Tax", we changed TAX to a different shape by using the left arrow to re-specify the storage area called TAX.

TAX ← +/ TAX

⌈ REDUCES TAX TO A SCALAR

If we use a comma between two arguments, what happens? It is called "catenate," the useful part of the word, concatenate. Let's save the answer, also.

```
ANSWER ← TAX , 1.00
⊞ ← ANSWER
0.725 1.000
2
```

Let's read it. "Answer is specified by tax catenated to 1.00" is the first statement. The second was "What is the shape of the display specified by answer?" First, the contents of the storage area called "ANSWER" was displayed on the third line and a 2 on the fourth line. The last display is the answer to the part of the question regarding the shape, it is now a 2 element vector. The dyadic comma joins items together into a vector. The following statement

```
⊞ ← PRICES ← PRICES, 34.21 29.95
0.37 6.34 27.98 48. 34.21 29.95
```

added two more prices to the list of 4 and saved the new list of 6 by specifying PRICES by the new vector.

The following APL statement applies the tax rate to each element of the list of prices, then adds the tax to the prices, and then adds them together by using sum reduction.

```
+/ PRICES + PRICES × TAX
157.496625
```

Stop and think about the previous statement. We have read many APL statements before and, of course, they are read from left to right. The English word statement above the last APL statement said that the tax rate was multiplied **and then** the tax was added to the prices, etc. If we read it from left to right, why did APL not also add up the prices first, then add the prices and then multiply the tax rate, in the same order? Why did it start on the right by multiplying the tax rate first?

The difference in the two possibilities is whether APL works from left to right or right to left. The last APL statement above would read "add the prices that are added to the prices multiplied by the tax rate."

Let's analyze this statement. It starts by saying: "Sum the prices."

What prices?

Those that are "added to the prices".

Which prices?

Those "multiplied by the tax rate".

If you think about it, you have to multiply the tax rate by the prices **first**, before you can add the amount of the tax to the prices and before you can sum them up to get the total. Yes, you have been tricked into starting at the **right**. APL executes statements from the right to the left. It uses an execution rule called the "right-to-left rule." Let's see how it works.

In mathematics, the statement:

5 × 7 + 3

could be executed as "five times seven and then add three, answer = (38)" using a left to right rule. On the other hand, it could be as "five times the result of seven plus three, answer = (50)" using a right to left rule.

In defining any computer language, the order of execution is a fundamental decision that must be made by the designer of the language. Most languages use a modified left to right rule. Usually

it is not just a simple left to right rule, but one that is modified by a set of rules that govern the order of execution of the primitive elements of the language. Many languages have a set of rules that read something like: "raise to powers first, multiply and divide next, then add and subtract," a three level hierarchy of operations.

Using parentheses is another way to force whichever order: $(5 \times 7) + 3$ or $5 \times (7 + 3)$, is intended. The parentheses override the rules. The most common language of the day when APL was conceived, FORTRAN, only had five primitive arithmetic operations and used the three level rule stated above. APL, however, has a very large set of primitive operations. Depending upon how you group them, the count is from 60 to 80 primitive operations. A set of rules to unscramble such a large number of possible hierarchies would be hopelessly complex.

Dr. Iverson devised the rule which has become known as the "right to left rule." In APL it is the only execution rule. That is it. It is simply: "Each function takes as its right argument everything to its right, to the right parenthesis of the pair that enclose it."

The ambiguous statement $5 \times 7 + 3$ is no longer ambiguous, the times sign takes the left argument 5 (since there is one) and takes *everything* to its right as its right argument. What is to its right? $7 + 3$. The evaluation of $7 + 3$ must take place before the right argument needed by the times can be used. Following this line of thinking, before the left most function can be executed, *everything* to its right must be evaluated. Likewise for all other functions in turn. The conclusion is that the right most function must therefore be evaluated *first*. Let's give the statement above to APL for evaluation:

$5 \times 7 + 3$

50

The result is 50 since the right most function + evaluates to 10 and the second step is 5×10 , yielding the answer 50.

The issue of hierarchy of operations in terms of the order of their execution simply ceases to exist. How elegant! It was a stroke of genius and represents only one of the creative contributions that exist in APL. Dr. Iverson probably used the criteria of simplicity to make such decisions. It did, however, bother a lot of folks who couldn't or wouldn't adjust to this new idea.

Alas, the rest of the world is moving in the opposite direction. AREXX, the AMIGA version of REXX, which has skyrocketed into prominence with great promises of extending the versatility of this amazing machine, uses a strict left to right rule except as modified by parentheses. It has an expanded set of 23 primitive operations and an eight level priority scheme for sorting out which operation operates when during execution. It is interesting to note that AREXX's priority 7, 6, and 5 is indeed the three level rule quoted above about exponentiation, then multiplication, division, and then addition and subtraction. The highest priority is Logical NOT and the lowest is Logical Exclusive OR. It is very likely that liberal use of parentheses will be found in AREXX programs in order to keep track of what is happening rather than relying on this complex hierarchy scheme.

APL also uses parentheses in the usual fashion. Because of the right-to-left execution rule, two interesting conclusions follow. If a right parenthesis is found at the end of a line, it is the right member of an unnecessary set. Further, if two right parentheses are found together, e.g. ")))", one of the sets is not required. As you write APL statements you will find that they can be written with a lot of parentheses or not, depending on your style. If you see APL statements written with a lot of unnecessary parentheses it isn't wrong, it simply indicates that the writer did not understand the right to left rule or



Great Prices! Shipping based on weight and zone.

For Information & Catalog Call:
Voice: 414-544-2066
Pursuable BBS: 414-544-6567

Spotlight on Hardware...

501 Clone, Spirit 512K	80.00
68030/4megs/882/80Q	2899.00
flickerFixer	460.00
Floppy Drive, SupraDrive	135.00
Floppy Drive, Internal 2000	90.00
Harddrive, Quantum 40	420.00
Harddrive, Quantum 80	625.00
Harddrive, Quantum 105	735.00
Harddrive, Supra 20M 500 ..	505.00
Memory, AdRAM 520 0k 500	120.00
Memory, Adv 2080 8Meg 0K	147.00
Memory, BaseBoard 0K 500	128.00
Meta4, TrumpCard Mem 0K	159.00
Modem, Aprotex 2400	85.00
Modem, Supra 2400	120.00
Mouse, Cordless	90.00
Mouse, Jin Mouse (Konyo)	50.00
SCSI Controller, Adv 2000	155.00
SCSI Controller, Byte/Sync ..	170.00
SCSI Controller, Word/Sync	170.00
Spirit Boards 0K	224.99
SuperCard AMI	72.99
SupraRAM 2000 0K	160.00
SupraRAM 2000 2 megs	315.00
Synchro-Express	55.00

Orders Only Please:
800-544-6599
Visa/MC/CODs

2414 Pendleton Place ■ Waukesha, WI 53188 ■ 9 AM to 5 PM M-F

Circle 134 on Reader Service card.

did not trust the APL interpreter to use the rule reliably.

Another trend in modern computing languages is tighter and tighter control over the "typing" of variables. In most languages a great deal of time and energy is spent making sure that the variables are integer storage spaces or are "floating point" (decimal) spaces, or variables that are able to take character data only, etc. You may have noticed that previous statements have freely mixed whole numbers and decimals. APL takes care of all such "typing" that is necessary. Since the storage is dynamically allocated at the time of use of the specification (\leftarrow) operation, the typing is simply determined by APL and becomes a part of the description of the variable, without any action on the part of the programmer. If there are no decimals, the array is automatically stored as integers taking much less storage than if there were decimals. Logical ones and zeros take one bit each, making maximum use of storage. If we create an integer matrix

```

□ ← MATRIX ← 3 5 p14
14 14 14 14 14
14 14 14 14 14
14 14 14 14 14

```

and then add one decimal value in the middle

```

□ ← MATRIX ← MATRIX[2;3] + .7
14.0 14.0 14.0 14.0 14.0
14.0 14.0 14.7 14.0 14.0
14.0 14.0 14.0 14.0 14.0

```

→

APL did a lot of work for us. The amount of storage required increased greatly after the conversion to the decimal notation. But why, Dr Iverson asked, should the user of a computer who wants to solve a mathematical problem be required to become a computer scientist in order to do so? Did you notice how easy it is to index the matrix at row 2, col 3 in the last APL statement above? ADA and MODULA2 are relatively new languages that require a lot of the programmer's attention because they have moved in the direction of tighter and tighter controls on the issues of types of variables.

Let's look at more of our primitive functions:

```
1 10
1 2 3 4 5 6 7 8 9 10
```

"IOTA" (ι) is the index generator and returns the integers from 1 to its integer right argument. Used dyadically, IOTA becomes an index finder and indicates the position that the right argument occupies in the left argument. For example, remember what we had at prices?

```
PRICES
0.37 6.34 27.98 48. 34.21 29.95
```

Did you remember that we catenated two values and respecified PRICES giving the six values? Of course you did. OK, for the dyadic use of IOTA

```
3 PRICES ι 27.98
```

indicates that 27.98 is the third element in the vector.

Here's another new primitive function:

```
8 ⌈ 7.3
```

"Ceiling" is the next higher integer above the decimal number right argument. Let's see if you can use this one in the reduction of a vector.

```
48.0 ⌈ / PRICES
```

and... hmmm. What did we do? Ceiling was extended by reduction to the vector PRICES. Recall what reduction does. It places the primitive in between each element of the vector and reduces the vector to a scalar. Let's peek inside and watch it in operation.

```
0.37 ⌈ 6.34 ⌈ 48.0 ⌈ 27.98 ⌈ 34.21 ⌈ 29.95
                27.98 ⌈ 34.21
                48.0 ⌈ 34.21
                6.34 ⌈ 48.0
0.37 ⌈ 48.0
48.0
```

As you can see the larger of each comparison between each pair in turn to each of the ceiling (or MAX in this dyadic use) operations, ending with the largest element in the vector. Try another variation:

```
7 9 4 ⌈ 7 9 2 ⌈ 5 3 4
```

which indicates why the statement was made earlier regarding "60 to 80 primitive operations, depending upon how you count them." The last three examples above of CEILING/MAX appear as different operations when in fact they are using the dyadic form of MAX in each case.

⌊ 7.3

7

"Floor" returns the next smaller integer. You should begin to have some ideas. Here is one possible idea: To round a vector of numbers (V1) type

```
V1 ← 2.3 6.8 4.5 5.0 5.4
V1 + .5
2 7 5 5 5
```

which would be read: "what is the floor of the numbers at V1 plus one half?" Did you see how it was done? You have to know what APL is doing and supply (think) the intermediate steps if you want to follow. With APL, however, at any time you do not understand what happened, you are free to put the problem in step by step to see how the answer was developed. For example:

```
V1 ← 2.3 6.8 4.5 5.0 5.4 ASET UP VECTOR V1
V1 + .5 A THE FIRST STEP, ADD .5
2.8 7.4 5.0 5.5 5.9
V1 + .5 AREPEAT THE FIRST STEP, TAKE FLOOR
2 7 5 5 5
```

Let's take a test to see if you are thinking APL yet. Let's think APL without giving it to the APL interpreter.

5 + 4 = 9

What do you think would happen? First, you were tempted to read 5 plus 4 equals 9, right? Now, If you also thought "yes, that's correct" you have to go directly to jail, cannot pass go, and you won't get out until you begin to think APL! In all fairness, you really needed another fact before you could have acted as a complete APL enthusiast. Let's give this statement to APL to evaluate. Before we do that, lets use APL in a discovery mode.

```
1 = 6
0
6 = 6
1
2 = 2 3 1 0 1 2 3 0 2
1 0 0 0 0 1 0 0 1
```

APL uses all the relational operators <, ≤, =, ≥, ≠, as propositions to be answered true (1) or false (0). So, back to our problem. One can always fire up APL and ask for the answer:

```
5 + 4 = 9
5
```

The result should be obvious. In case it isn't, do it one element at a time:

```
4 = 9
0 A THE ANSWER IS NO, 4 IS NOT EQUAL TO 9
5 + 0 A 5 PLUS THE RESULT OF THE FIRST STEP, 0
5
```

Let's move along and try another primitive:

```
? 100
34 ? 100
3 ? 100 100 100 100 100
67 0 49 71 67
```

This one might take a little time for you to figure out just what is going on. Random numbers are being drawn from each element of the right argument. The repeat of 67 indicates that the random draws are with replacement so that each draw from each vector

element is independent of all others. OK... do you suppose it can be used diadically? Try it.

```
10 ? 100
4 6 54 5 97 84 9 10 66 23
```

Hmmm... Suppose...

```
10 ? 10
4 7 3 8 2 1 9 5 6 10
10 ? 10
8 1 10 3 5 6 2 9 4 7
```

Yep, it gives random numbers, the left argument tells how many and the right argument gives the domain. Ten random numbers from the domain 1 to 10 (10 ? 10) gives all of them in random order! A random permutation. The first request asked for only ten of the 100 numbers in the domain 1 to 100. Neat huh? The diadic use is called "deal," the monadic use is "random."

Yes, you can try it:

```
100 ? 10
DOMAIN ERROR
100 ? 10
^
```

Despite all the neat things APL can do, it can't get more out of mathematics than there is in the definitions. Let's try one more:

```
DATA ← ? 10000 p 1000
```

What did we do? Read it. "Data is specified by random draws from 10,000 reshaped 1,000's." If it sounds as if we independently drew 10,000 random numbers, each from the domain 1 to 1,000 and stored them away at DATA, you are right on track. You might check it out.

```
DATA
```

WOW, are you sure? It will display 10,000 random numbers! Go ahead and try it on the AMIGA. The editor, however, wouldn't want to do it here in this article! On the A2000 it takes about 49.66 seconds to display them at 14 across in each row. Maybe we better check before asking for this display.

```
p DATA AREQUEST THE SIZE OF DATA
10000 ASURE ENOUGH! IT WOULD HAVE SHOWN ALL 10,000
DATA [ 15 ] ASHOW THE FIRST FIVE ELEMENTS
134 6 689 942 134
```

Yes, you can use operations such as IOTA to generate the numbers for indexing the array DATA.

The best way to learn about APL is to load the APL interpreter, click on the APL icon and try things out. If the operation is legal it will give the answer, if not it may be a DOMAIN ERROR or some other indication that the operation is not defined on every kind of number or structure in existence. OK, you have been eager to try

```
5 + 0 AREAD "5 DIVIDED BY ZERO"
DOMAIN ERROR
5 + 0
^
```

but it indicates that dividing by zero is outside the domain of the mathematician's definition of division. Just what your mathematics teacher always told you. Neat way to learn mathematics. As a matter of fact, learning APL would be a whole heck of a lot more productive in contributing to a child's education than learning BASIC. By learning BASIC you learn all kinds of arbitrary rules,

CONTROL THE UNIVERSE!

"I'm totally awed by what you have done!...it's beautiful, especially when the lights are off...congratulations..."

Arthur C. Clarke

author of 2001: A Space Odyssey

Distant Suns, the award-winning planetarium program, is endless entertainment and education for all. \$69.95 retail. Ask for it!



Virtual Reality Laboratories, Inc.

2341 Ganador Court
San Luis Obispo, CA 93401 USA

Circle 121 on Reader Service card.

unique to BASIC, in order to learn computer literacy and how to write programs for a computer. By learning APL, you learn all those things and have a discovery-based mathematics workshop too. Which is more productive will have to be judged by each individual.

Next time we will do some statistical calculations and begin to look at the way that APL is implemented on the AMIGA. Now that you have seen that APL is a friendly language and easy to use, it is time to show some of the power of its implementation on the AMIGA.

ABOUT THE AUTHOR

Dr. Henry T. Lippert is an educator. He has specialized in the application of computers in education and training and to the tasks of the instructor and the instructional designer. He was one of the original developers of Computer-Based Instruction (CBI) at the University of Illinois during the 1960's. He is the Chief of the Instructional Methods Division at the Academy of Health Sciences in San Antonio, TX.

APL INTERPRETER FOR THE AMIGA
Not copy-protected. Includes reference manuals.
Spencer Organization, Inc.
24 Wampum Road
Park Ridge, NJ 07656
(201) 307-9099
Inquiry #203

•AC•

PD Serendipity

*Insight into
the World
of Public
Domain
Software
for the
Amiga®*

HIGHLIGHTS OF THE MOST RECENT ADDITIONS TO THE FRED Fish collection include virus checkers, a simple encoder program, and a 200-page C Manual with more than 70 executable examples. These programs can be found on disks 331—340.

VIRUS CHECKERS

The first virus checker I came across is found on FFD 331, **LVR** V1.20. This program detects and removes link viruses from disk, among them IRQ, BSG9 (TTVI), Revenge of the Lamer Exterminator (a fairly new virus) and the Xeno virus.

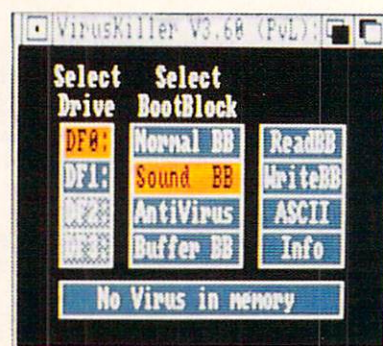
LVR can either search a directory or an individual file for a known virus. LVR scrolls the names of the files it is checking, but unlike most virus checkers I have come across, it does not indicate the number of files checked when finished. *Author: Pieter Van Leuwen*

VirusUtils can be found on FFD 331. VirusUtils contains two virus checkers, **VirusHunter** and **VirusKiller**. VirusHunter removes all known viruses from memory. Some known viruses include the SCA (and clones), Revenge, ByteBandit and ByteWarrior. For a complete list, check the document file in the VirusUtils directory on FFD 331.

VirusHunter does not scroll files when checking for known viruses; however, it does display a message ("No known virus found", for example) upon completion.

VirusKiller V3.60 removes all known viruses from memory and disk. After removing the viruses, the disks can be checked without the viruses copying themselves back to the disks.

When activated, VirusKiller displays a new window on the screen. Here you give instructions to read the bootblock (readBB), write to the bootblock (writeBB), select the type of bootblock, and select which drive to check.



VirusKiller V3.60's virus checker window.

by Aimée B. Abren

At the bottom of the VirusKiller window is a long rectangular box which displays messages to the user, such as: "disk removed", "no disk in drive", "disk contains normal BB", and "disk inserted".

VirusKiller destroys SCA, Revenge, ByteBandit(2) and Turk, plus others. For the complete list, check the document file on FFD 331. *Author: Pieter Van Leuven*

PROGRAMS

Kryptor V1.0 (FFD332) is a simple file encoder/decoder. The program is easy to use once you get the procedure down. An encoder is a program that allows you to turn text files into unreadable "garbage" characters for privacy or protection. These files are password protected, and can only be decoded with the correct password. Once decoded, the files are again readable text files.

The document files come in both German and English. Unfortunately, the requestor boxes are all in German. This makes it a little difficult when the requestor gives you two choices. After experimenting for a little while, you will learn what responses give you the results you want.

Encoding is as simple as selecting the file you want encoded, typing in a password, and closing the window. If you do not click the window closed, you will be prompted to type in another password. This loop will continue until you click the window closed, allowing the use of more than one password for a file.

Once encoded, your file will be printed as garbage on the screen until the file is decoded. To decode your file, simply repeat the steps to encode. *Author: Michael Balzer*

PPMore V1.5 (FFD334) is a program that reads ASCII files and files crunched with PowerPacker.

You've probably heard of a similar program called *More*. PPMore is different because it uses gadgets and loads the whole file in memory, while *More* loads only sections at a time.

When loaded, PPMore displays a new window. The filename and number of lines in the file are located in the title bar. The bottom of the screen displays the following options: Next Page, Prev Page, Next Line, Prev Line, goto N%, Print File, and Exit PPMore. They do as their names imply.

As mentioned earlier, PPMore reads files crunched with PowerPacker, a program used to compress files for more disk space. In the document file of PPMore, it claims to be able to read encrypted files as well. If a file is encrypted, a requester will appear asking for the password of the file. *Author: Nico Francios*. (Look for PPSHOW, which also was written to compliment PowerPacker. PPSHOW is similar to *Show*, in that it displays IFF files. It can be found on FFD 334. *Author: Nico Francios*.)

ANIMATION

AniPtrs (FFD332) is a collection of several animated pointers from fish to cats to space ships. These animated pointers will dress up any Amiga screen.

Images were created with DeluxePaint III and Pointer Animator was used to compile the animations. *Author: Bob McKain*

GAMES

Your task is to clean up the viruses in your SYSOP's hard disk in the game **SYS** (FFD 336). These creatures have infected every sector of your disk and it's your job to get rid of them.

To kill a virus you simply kick a disk, 5-1/4" or 3-1/2", at it. But watch your back—the higher the level, the trickier they are.

Each virus has its own personality. The mouse, who is quick on those corners, and the joystick are the milder viruses. The photo diode is sneaky, so proceed ever so cautiously. The magnet is the most dangerous virus. He'll follow you everywhere, especially on the higher levels.

While chasing these creatures thru your hard disk, all is not glum. There are bonus surprise boxes where you can grab some extra points and gain another life.

The game can be controlled by the numeric keyboard or joystick. To pause the game, move the cursor to the right side of the screen. Some function keys are used as switches to turn music on (F6) or off (F7), to skip to level 10 (F10), etc. *Author: Anders Bjerin*

MANUAL

Included on FFD 336 is a 200-page **Amiga C Manual** with more than 70 executable examples, including source. Some topics discussed include working with screens, windows, graphics, gadgets, requesters, alerts, sprites, etc.

Examples are written in Lattice C V5.02 but will compile with older versions as well as with other C compilers.

Registered users can update for the cost of a new disk and postage and handling. *Author: Anders Bjerin*

EXTRA, TWO!

Two programs that seem worth taking a peek at are **IFF2Ex V1.0** (FFD331) and **SID V1.06** (FFD338). IFF2Ex converts IFF pictures to an executable file. *Author: Peter Van Leuven*. SID is a directory utility for the Amiga. *Author: Tim Martin*

UPDATES

- **Csh V4.01a** (FFD331) is an update to V4.00a on FFD 207. This program is an alternative command interface including command line editing, aliases, piping and more. Changes include restoring window title after quitting, and long lines in source files can be broken into more than one line. *Authors: Matt Dillon, Steve Drew, Carlo Borreo and Cesare Dieni*

- **Multiplot V XLNb** (FFD333) is an update to V XLN on FFD 292. Multiplot is a program for making 2D plots. Bug fixes permit the handling of greater than 20 data sets and reset plot dimensions when a plot is redrawn without text. New features include the custom plot window allows selection of different plot symbols, and all error reports are displayed by alerts or requesters; no error messages will appear in the current CLI. *Authors: Alan Baxter, Tim Monney, Rich Campeaux and Jim Miller*

- **C++** (FFD338) is an update to C++ on FFD 28. This program is an implementation of the C coprocessor. ANSI features were added to this updated version. Version number not known. Source code included. *Authors: Martin Miow and Olaf Serbert*

- **PCQ V 1.1c** (FFD339) is an update to V1.0 on FFD 183. PCQ is a sub-set Pascal compiler. This program includes compiler source and example programs. *Author: Patrick Quaid*

- **Plplot V 2.6** (FFD 340) is an update to V1.00 on FFD 222. Plplot is a library of C functions useful for scientific plotting on the Amiga. Some fixes include an improved intuition interface, new device drivers and the capability of adding additional device drivers easily. *Author: Tony Richardson*



R O O M E R S

by The Bandito

[The statements and projections presented in "Roomers" are rumors in the purest sense. The bits of information are gathered by a third party source from whispers inside the industry. At press time, they remain unconfirmed and are printed for entertainment value only. Accordingly, the staff and associates of Amazing Computing™ cannot be held responsible for the reports made in this column.]

THERE'S PLENTY OF COMMODORE news this time. Seems that the new administration has gotten quite a few pots boiling, and the soup will be served this year. The Bandito has chased down a few hearty chunks of data, so sit back while the data stew is served.

Atari rushed out their STe computers in the fall before the software was quite ready, and as a result few ST programs were able to run on the new machines until Atari put out a software patch. Even now, there is still a lot of software that does not work on the new boxes.

The latest scuttlebutt is that the launch of the 3000 will highlight AmigaVision, the new "multimedia" software from Commodore that combines the best features of UltraCard, CanDo, and DeluxeVideo. Well, the Bandito isn't sure just how whizbang AmigaVision will really be; you'll have to wait and see for yourself.

Some developers are annoyed that Commodore is publishing such an important piece of software and cutting into their business. Especially since the latest data heard by the Bandito is that AmigaVision may well be bundled with all Amigas sold. This has publishers of certain products running a bit hot about competing with a piece of free software. On the other hand, bundling AmigaVision could help sell quite a few more Amigas, which means publishers can sell more software.

Commodore stock is down below 8 these days, lower than it has been for years. But computer stocks have generally been off. The stock could revive if the Amiga

3000 is well received. A lot depends on Commodore's ability to generate Amiga sales in the United States.

The chance for an explosive Amiga market growth is still there, but hopes grow more distant as Commodore's big effort over Christmas failed to ignite sales. On the other hand, new developments offer hope once more to those who still have not given up the Amiga Dream. The Amiga 3000, the Video Toaster, the Amiga CD-ROM project, and AmigaVision have all promised to supercharge Amiga sales. The faithful Amiga fans are still hanging in there.

The Amiga has never become a smash seller, but it has demonstrated staying power. Commodore seems to be the only computer manufacturer left that is even trying to go after the home market. The Bandito thinks the A500 is more important than the A3000, because the potential market is huge. The success of the C64 could happen all over again if the Amiga finds that magic price point.

The Bandito has his informants overseas, and they report more news on Commodore's CD-ROM machine. The word is that working prototypes have been shown to major developers in the UK. It's said to be an Amiga 500 motherboard with a built-in CD-ROM drive and the full complement of standard Amiga ports. Pricing is not fixed yet, but the target is said to be under \$750; depending on component costs, perhaps it could be as low as \$600 at introduction.

Game developers hope and pray for CD based entertainment, and Commodore's widget could be the answer to their prayers. Imagine games with full CD quality sound tracks, gobs of animation, and great sound effects—it would be super! Commodore's Amiga CD solution looks good on paper—it has the lowest initial price point, which is crucial for a machine that would attract game players. It's still more expensive than people are really willing to pay for a game machine; hopefully, the price can come down fast enough to guarantee success. The initial Amiga

software base will help sales. The development tools are mature, so new software can be developed quickly.

But Commodore is still contemplating its low end product strategy, and no decisions are set in stone yet. The pricing of the CD-ROM Amiga will be around \$750. The Amiga 500 is dropping to \$499 by this Christmas. And they have prototypes of a console machine (the Amiga 250?) with the chips of an Amiga and plenty of ports (including a cartridge port) but no keyboard or disk drive, which they could sell for under \$250. The problem is devising a clear marketing strategy for these "similar-yet-different" products. They all have their good points and bad points. The worst thing would be to throw them all on the market with no marketing support. The Bandito thinks they should call an Amiga-based game machine something other than an "Amiga", if they want to sell A3000's. After all, would you buy a Nintendo business workstation?

It will take some time for the dust to settle, but the Bandito predicts that September will be a busy month for Commodore product introductions.

FISHING FOR NETWORKS

Networking is finally moving onto Amigas, and it looks as though 1990 is the year that Amigas will finally be able to hook up to the rest of the computing world. At a recent Sun networking show, it was revealed that Commodore has tested a version of Sun's Network File System (NFS) for the Amiga. But no versions of NFS for OS/2 or Macintosh were shown. Could it be our friend the Amiga taking the lead?

A company called Hydra Systems has an AmigaNet Ethernet/Cheapernet card at a list price of about \$700; it was demonstrated at AmiEXPO. The card also supports TCP/IP, for those who know about that stuff. And the Bandito hears whispers that an unnamed Amiga developer (whose initials are GVP) may be working on a similar card, though at a lower price. And Commodore has an Arcnet card in the works, for possible release this summer. CMI's Ap-

pletalk card may yet find its way to the marketplace. Even ASDG has announced they're working on a networking card. And Commodore has been talking to Novell (among others) about software support for Amigas on their popular networking systems.

WHEN IT RAINS IT POURS

As if Mediagenic has not had enough troubles lately (losing a patent suit with Motorola that cost them millions), they've got another lawsuit on their hands. Mediagenic bought ZSoft (makers of the bestselling PC Paintbrush) in 1988 for some of their stock. Now ZSoft's president claims that the stock is only worth a fraction of what Mediagenic said, and that it has not declared any dividends (and the stock price has dropped). Meanwhile, ZSoft stock made a profit of \$700,000, which all went to Mediagenic. So ZSoft wants control of their own company (and profits) again.

While there's trouble at Mediagenic, things are looking bright for Electronic Arts. They reported an increase in profits and sales for 1989. And Electronic Arts is now an official Nintendo licensee, thus putting to rest their efforts to reverse engineer the Nintendo carts. The Bandito supposes that it was cheaper to "join 'em than fight 'em," as the saying goes. Nintendo is where the money is being made, so this should help Electronic Arts' sales and profits.

Amiga entertainment software is being developed in Electronic Arts' UK division, since the Amiga is the #1 computer in Europe. Electronic Arts is also avoiding roleplaying games unless they can import them. Roleplaying games tend to take a long time to develop, and the payback is chancier. Then again, the big hits are great moneymakers.

While we're talking software, the Mouse is planning a frontal assault on the entertainment and creativity business in the Amiga market. Disney apparently plans a stronger presence in the software business, showing their muscle recently by hiring Accolade's marketing veep and a former Electronic Arts producer to run their operation. Apparently, the mouse folk are serious about making money on software. Their new animation package should carve itself a good niche in the market. Disney apparently hopes to imitate LucasFilm's success in turning movie titles into hit software.

The latest Software Publisher's association figures are in for 1989. For entertainment publishers, the news was not so good. The overall market shrank by 12%. While IBM and Amiga software sales grew (and Amiga grew the most), the C64 and Apple II markets took a nosedive. Amiga is now recognized as the #2 target market for entertainment software development in the

U.S. In Europe, Amiga is #1 for entertainment. Electronic Arts has a number of Amiga titles in development at their UK subsidiary, so look for most of their Amiga action to come from overseas.

NINTENDO UPDATE

There is some news to report in the Nintendo/Tengen court battle. Give this round to Nintendo, for they won the right to sue retailers who buy Tengen cartridges. While the lawsuits won't really be resolved for years, this is an important victory for Nintendo. You can bet it had some influence on Electronic Arts' decision to sign a licensing agreement. The other big news in Nintendo-land: Nintendo has told its licensees that they can now get as many cartridges as they care to order. This solves one of the biggest problems for licensees: since Nintendo was the only manufacturer of cartridges, there was always a shortage. With the availability problem apparently at an end, publishers can go full speed ahead.

So the Nintendo juggernaut keeps on rolling, though the graphics are looking pretty old. The Genesis machine looks very much like an Amiga, and so far it seems to be selling all right. Does the Amiga game machine have a shot? The Bandito thinks that Commodore may well decide to forget about it and concentrate on their CD-ROM device instead, even though it is much more expensive.

DCTV is the latest word buzzing around Amigadom. Supposedly, Digital Creations has a low cost way to expand the Amiga's palette with some sort of frame buffer that will sell for around \$500. But they are not the only one, and Digital is not exactly rushing to the marketplace. Whatever happened to the V-Machine, which was supposed to be a Toaster-Killer? Looks like it won't be out for a while, and when it does happen it will be considerably more expensive and less effective than the Toaster. However, the V-Machine will work with Macintosh or IBM. Once again, owners of other computers can expect to pay more to get what the Amiga gives you for less.

SOME JAM WITH THAT?

It is finally getting real, sports fans. The long-awaited Video Toaster was the smash hit at the National Association of Broadcasters (NAB) show in Atlanta. People were stacked up like airplanes at O'Hare waiting to see the ultra-cool Penn & Teller videotape, and live demos of the Toaster's effects (by their new media star, Kiki). The product is in FCC approval now, so volume shipments will begin when the Feds finish (probably in the summer). Jaws were dropping all over the convention, and the Video Toaster won the award for Best New Product. The Toaster was also fea-

tured in CNN reports on the show. The Bandito thinks the most interesting part was the reaction of industry types. They realized that the Amiga + Toaster combination may be able to destroy the market for their current zillion-dollar equipment.

Makers of inexpensive effects boxes (which used to cost \$20-30,000) can see their doom on the monitors. The Bandito predicts that sales of other DVE's will come to a crashing halt in anticipation of the Video Toaster's imminent release. Scuttlebutt says that NewTek has presold more Video Toasters than all the other DVE's ever sold before. Commodore, of course, is breathlessly awaiting the Video Toaster's impact on Amiga sales. Should move a lot of A2000's in anticipation, says the Bandito.

No one was neutral; they either loved NewTek or hated them. Among the hatees: Apple Computer, which saw its efforts to promote the Macintosh as video machine wholeheartedly ignored. On the flip side, all the users of video equipment could see the benefits of this new technology, and they were clamoring to get their hands on the hardware.

SHOWSTOPPERS

GVP, in an impressive booth, showed off their A4000 plus 1, a 40 MHz 68030 accelerator card that is a screamer. They're planning a 50 MHz version for summer release. And when Motorola makes the 68040 more widely available, you can bet GVP will have a board out.

Showed off was the Rejuvenator, a board to upgrade the A1000 to the A2000 level, able to use the new Enhanced Chip Set and even a flickerFixer (they have added a video slot that is almost the same as the A2000 video slot). Their booth was crowded, and they sold quite a few Rejuvenators to eager A1000 owners looking to bring their machine up-to-date.

On the other hand, a company called Delaware Valley Software announced a board similar to the Rejuvenator (the DVS Wonder), but they did not show a working version. Just a few flyers, which means nothing when someone else is actually selling a product.

The World Of Amiga show in New York was a bit of a turnaround for Commodore. Previously, they insisted that the show be called the World Of Commodore and used the opportunity to show off C64's and Commodore PC clones as well as Amigas. After the fiasco in Los Angeles last year, where they had fewer attendees than an Atari ST dealer's convention, Commodore wised up. The only reason for a Commodore show is the Amiga. No one cares about the C64 anymore, and if you have a PC clone, you go to a PC clone show

The Command Line

by Rich Falconburg

THE VARIETY OF SYSTEMS IN THIS WORLD OF computers is astounding. How can you possibly choose the type of hardware you want and still find software that will run on a particular machine that will let you accomplish the job you wish to do? For many, this is not a question of great importance. Just pick the greatest computer (the Amiga, of course) and buy it. For some, however, the purchase of a system to use at home is often driven by requirements of a daily job. Will I be able to finish work at home if I need to? Are the programs I am comfortable with and use on a daily basis available on a machine that I would like to own? All too often, the answer is a resounding NO. Inevitably, you end up making some sort of compromise and learning to live with the consequences. Occasionally, a piece of software might actually be file compatible with a program you must use at work. In even rarer cases, you will have an identical program written for your specific computer. Obviously, the preference would be the last choice, especially if the data files are compatible. The program we'll look at this month falls somewhere in-between the last two descriptions. It provides an interface that retains a format many will be familiar with. In many cases, it performs equally as well as the program it is designed after. Nevertheless, it does fall short in certain areas, requiring some adjustments.

In my daily usage of various operating systems, I find that it becomes tedious to constantly try to remember which system I'm using so that I enter the correct commands for that system. Through the use of Aliases, both at work and on my Amiga, I've been able to alleviate much of the headache. But that is only a stopgap measure. This becomes obvious when I accidentally try to perform an operation specific to a given environment. I have yet to find a machine that doesn't complain when you don't speak its language. Very intolerant, these technological beasts! Ah well, artificial intelligence is in the making, and some day it will help make up for my bungling. Until then, I must learn to adapt or find ways to teach my old dog some new tricks.

With the help of programmers such as Steve Koren, I am doing just that. He has provided the Amiga community with a command interface that is very similar to the UNIX Korn shell (interesting play on names, eh?). His SKsh shell is an environment that not only provides most of the UNIX commands that I use daily, but also gives me much of the same command line functionality that I've grown so accustomed to. This package offers some excellent features, for both the casual and the power user.

The tremendous effort put into this Freely Distributable bundle of software speaks of someone with a flavor for professionalism. Steve

went to great lengths to produce a package that in my opinion rivals the quality of those produced by some commercial software houses. The extensive documentation is clearly written and well organized. Included are a User's Guide and a detailed Reference Manual, both impressive in themselves. The User's Guide is packed full of information and examples covering the operation of SKsh. In the 100-page Reference Manual, he explains in detail the operation of every command, complete with short examples, usage templates, and cross references. Each includes a Table of Contents, and the User's Guide includes an Index. Other documents cover installation, error codes, differences between SKsh and the Unix ksh, hints on using the shell more effectively, bug reports, limitations, technical information, update addendums and more. All in all, it is a very complete package. And each page is nicely formatted to print on a standard page. The following is an overview of what this shell does.

The built-in commands include:

alias	set or examine aliases
argcount	obtain a count of the arguments supplied
basename	print the base (file) name of a given path
break	break from an enclosing for or while loop
cat	copy files or stdin to stdout (similar to TYPE)
cd	change the current working dir (similar to CD)
chmod	change file permissions (similar to PROTECT)
continue	resume next iteration of loop
cp	copy files (similar to COPY)
date	print date and time (similar to DATE)
dec	dec 1st argument by 1 or 2nd argument
dirname	print directory name of a given path
echo	echo arguments (similar to ECHO)
exit	leave sksh
export	set value of an AmigaDOS env: variable
extname	print extension part of file spec
false	do nothing; return false
getenv	get value of an AmigaDOS env: variable (similar to GETENV)
history	list command history
inc	increment 1st argument by 1 or 2nd argument
info	get device information (similar to INFO)
local	make local variables
ls	list directories (similar to LIST)
match	print arguments matching first pattern
mem	list system free memory (similar to AVAIL)
mkdir	create directories (similar to MAKEDIR)
mv	move files or directories (similar to RENAME)
options	set or examine sksh option flags
ps	get process status information (similar to STATUS)
read	set a variable value from standard input
return	return from function or script
rm	remove a file or set of files (similar to DELETE)
rmdir	remove an empty directory
set	list vars, aliases, builtins, or funcs
shift	shift arguments to function or script
sksh	the main executable for the shell
sleep	delay for n seconds
source	execute a file in current context
touch	update modification times on files
true	do nothing; return true
unset	remove a variable
version	print sksh version identification
whence	print info on interpretation of name
which	print the full pathname of file (similar to WHICH)

Shell script keywords include:

expr: A type of calculator with embellishments. This command is useful for extracting or matching strings.

for..do..done: Executes a statement list in order

force: Forces the shell to interpret the given command as one of a function, a builtin, or an alias.

function: A function is similar to a C function and allows you to establish a command that executes a series of statements to perform its operation.

if..then..elif..else..fi: This is the standard testing mechanism used in most programs.

time: Displays the execution time, in seconds, of a given command.

while..do..done: The standard looping construct used to perform a sequence of operation based on a set of conditions.

Environment variables used by the shell:

SKSHINIT: Two files are included, .skshinit and .skshrc, which get executed at each invocation of the shell. The SKSHINIT variable points to the user configuration file .skshrc. This may be altered to point to a different file.

CLINUM: This variable contains the CLI process number of the current shell.

CMDNUM: This variable increments each time a command is entered.

COLUMNS and LINES: These two variables contain the height and width of the current window. Only the COLUMNS variable is currently used and will allow SKsh to adjust the line length if the size of the window is changed.

HISTSIZE: This variable contains the size of the this history buffer.

HOME: This variable defines the default destination for the cd command, that is, when cd is entered by itself with no directory specification it will check this variable and change to the directory that it finds there.

IFS: Or Internal Field Separator. This character is used to break text from variables and command substitution into separate parameters.

LOGOUT: This variable is executed when the exit command is used to quit the shell.

OLDPWD: This variable contains the previous directory. The -p switch may be used with the cd command to return to this directory.

PATH: This variable contains a list of the available paths.

PNPC: This is a special variable used by SKsh to define the number of non-printing characters in the current prompt string.

PS1 and PS2: Each of these contains the primary and secondary prompt strings, respectively.

PWD: This variable contains the current working directory.

SHELL: Set initially to "sksh", this variable may be used to in shell



MOVING?



SUBSCRIPTION PROBLEMS?

Please don't forget to let us know. If you are having a problem with your subscription or if you are planning to move, please write to:

Amazing Computing Subscription Questions
PiM Publications, Inc.
P.O. Box 869
Fall River, MA 02722

Please remember, we cannot mail your magazine to you if we do not know where you are.

Please allow four to six weeks for processing.

scripts to establish execution criteria. Although not 100% compatible with the UNIX ksh, some ksh scripts may run under SKsh and limitations can be defined using this variable.

SYSNAME: This variable may also be useful as described above. Currently SKsh only runs on the Amiga and such is the contents of this variable.

In addition to the commands described above, several aliases are defined in the .sksinif file which has the affect of adding several useful commands to the collection. The command line editor is fashioned after the EMACS editor, thereby making this shell useful via other devices such as a simple Alpha-numeric terminal. I've experimented with this approach using AUX: and have been very happy with the results. We'll examine that more fully next issue.

This command shell has proven to be very reliable. The only difficulty I ran into was covered very clearly in the documentation. If you intend to use this shell, be sure to get a copy of the latest version of arp-library (1.3 version 39) and save yourself a lot of headache. I had an earlier version and had trouble getting SKsh to parse AmigaDOS commands properly.

Next issue I'll begin a short series on how to turn your Amiga into a multi-user powerhouse through some more programs available in the public domain.

•AC•

Send questions or comments to Rich Falconburg, c/o *Amazing Computing*, P.O. Box 869, Fall River, MA 02722-0869, or send Email to R.Falconburg on GENie.

List Of Advertisers

Need more information?

Need help?

Need to know?

Contact the AC advertisers!

Please use the Reader Service Card to contact those advertisers who have sparked your interest. Advertisers want to hear from you. This is the best way they have of determining the Amiga community's interests and needs. Take a moment and contact the companies with products you want to know more about. And, if you wish to contact an advertiser directly, please tell them you saw their advertisement in

Amazing Computing
For The Commodore Amiga

Advertiser	Page	Reader Service Number
ACDA Corporation	17	104
AmiEXPO	CIII	119
AmiTech Computers	45	114
Beta Unlimited	50	126
Computability	25	117
Computers Etc.	56	113
Delphi Noetic	67	142
Digisoft	44	137
Fairbrother & SoeperMann	34	179
Grapevine Group Inc., The	12	147
Grass Roots		
Video Productions	34	108
Hologramophone Research	35	109
Hunter Group, The	9	111
Impulse, Inc.	5	115
Imtronics	CIV	151
Inovatronics	7	110
InterComputing, Inc.	11	116
Krueger Company, The	66	118
Lattice Inc.	33	124
Memory Location, The	63	107
Memory Location, The	34	186
Micro Miga	46	182
MJ Systems	62	149
One Byte	61	135
Puzzle Factory, The	3	168
S.P.O.C.	13	112
Safe Harbour Software	71	134
Saxman Systems	43	103
Silver Fox Software	54	105
Straightline Software	28	106
Twilight Games	55	175
Virtual Reality Laboratories	73	121

AMIGA 3000 Launch

Multimedia and AmigaVision Take Center Stage



New York, April 24, 1990. In the dark, historic Palladium, the Amiga 3000 made its debut in front of over 350 members of the press, Amiga dealers, developers, and a host of VIP's. The event not only sponsored Commodore's newest addition to Amiga line, but established CBM's hold on the multimedia market. Preceded by a "countdown" flashed on overhead screens, Commodore introduced multimedia via demonstration with a superbly produced multimedia video created entirely on the Amiga. From Professor Corey, to a space launch, to the moon, and back to the past, CBM pulled all aspects of multimedia together for one powerful punch.

CBM's President and CEO, Harold D. Copperman, was instrumental in generating Commodore's and the Amiga's predominance of multimedia stating, "Today's event—and, later this week, World of Amiga—give us an opportunity to re-introduce you to the Amiga technology. It's a technology positioned to take advantage of the hottest marketing opportunity of the 90's: Multimedia.

"It's our turn, it's our time. We had the technology 4-1/2 years ago. In fact, we had the product ready for multimedia before multimedia was ready for a product. Today we're improving the technology ...and we're in the catbird seat. It is our time. It is Commodore's time."

Mr. Copperman went on to say, "...I'm at Commodore just as multimedia becomes the most important item in the marketplace. Once again I'm with the leader. Of course, in this industry a leader doesn't have any followers; he just has a lot of other companies trying to pass him by! But take a close look: the other companies are talking multimedia...but they're not doing it. They're a long way behind Commodore—not even close."

Mr. Copperman later described multimedia. "Multimedia is a first-class way for conveying a message—because it takes the strength of the intellectual content... and adds the verve—the emotion-grabbing, head-turning, pulse-raising



Commodore's Cast: Amiga developers and Commodore executives teamed together to create the multimedia event.

impact that comes from great visuals plus a dynamic sound track. For everyone with a message to deliver, it unleashes extraordinary ability. For the businessman, educator, or government manager, it turns any ordinary meeting into an experience."

The Amiga 3000

After listing the benefits and needs for multimedia, Mr. Copperman introduced the A3000. (For details on the A3000 see the May issue of *Amazing Computing*.) As part of the full effect of a multimedia presentation, Commodore used prepared video tapes of Jeff Porter, Commodore's Director of Engineering, and Walt Simpson, Director of Product Marketing, to discuss the A3000.

The Amiga 3000, starting at \$3299 for the 16MHz 40MB hard drive system, boasts a long list of special features. New designs in its hardware allow the A3000 to advance to a higher level of hardware and software technology while remaining compatible with older versions. Its new display and operating system not only improve the look of the screen but provide the Amiga user with a new set of tools.

AmigaVision™ Announced

AmigaVision, CBM's hot new authoring system for the Amiga, will now be bundled with all new Amiga 2000 and 3000 systems sold, according to CBM Director of Marketing, C. Lloyd Mahaffey. Mr. Mahaffey said that a special offer would be available to all Amiga owners who purchased their computers within 90 days of the announcement.

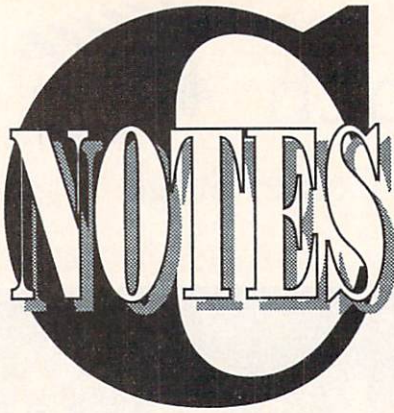
The icon-driven, graphically oriented AmigaVision is Commodore's key device to place multimedia at the home user level. David Archambault, CBM's director of business markets, said, "Unlike other authoring systems, AmigaVision is a non-intrusive environment which allows applications to run without the AmigaVision menus being visible. Enhanced design capabilities allow for immediate on-screen preview while the presentation program is running. This allows the user to create unique user interface elements quickly and with ease."

Mr. Mahaffey went on to say, "AmigaVision is a key component of Commodore's product marketing strategy for the Amiga family. While professional users comprise the majority of the multimedia related markets today, future plans call for penetration into the consumer market as home users begin to discover the benefits of multimedia."

Multimedia: Commodore's Territory

There is little doubt that Commodore is going to continue to push their advantage in the multimedia market. The event they staged at the Palladium, as well as their presence at World of Amiga (please see the related article on page 47 of this issue), has demonstrated Commodore's commitment to maintain their advantage. With tools such as the A3000, the new Workbench 2.0, and AmigaVision, CBM has taken more than a lead over their competition. They have staked their claim on multimedia with an open challenge to all.

•AC•



From the C Group

by Stephen Kemp

IF YOU'VE BEEN IN COMPUTER PROGRAMMING A while you have probably heard the terms "linked list" and "doubly linked list", and you may have read one or more of the numerous articles that have been published on these subjects. Of the articles that I have encountered, the author usually spends lots of time covering the theories and definitions of doubly linked list technology without spending much time on implementation. Rather than reinvent the wheel, I'll spend more time on actual examples. Some basic definitions are in order, however, for those who may not be familiar with linked lists and as a refresher course for everyone else.

Linked Lists: Data location made simple.

A linked list is a method of storing information (data) in such a way that one data item "points" to the item that logically follows it. In other words, if you know where one member of the list is stored, it is possible to find all the elements that follow it. All the items are "linked" to each other. A doubly linked list follows the same principle, but has the added capability of "pointing" to items that occur prior to the current member. This means that you can traverse a list forwards and backwards.

Since the items in the list are linked to each other, it is easy to have any number of items in the list (often called a queue). This means you don't have to know ahead of time, as with arrays, how much data space to reserve for the list. Making room for a new item in a queue is as simple as calling a memory allocation (malloc) function and setting the pointers to maintain the links. The implementation that I'll be discussing uses doubly linked lists.

The key things to know about a queue is where the first and last elements of the list are located and how to reach the member that occurs before and after any particular element. The easiest way to handle these tasks is through structures. Listing One is the "include" file that is used by the subsequent code. In it is defined the prototypes of all the functions, as well as two key structures.

The first structure type is named QITEM. Given any member, you must be able to find the previous member ("prev") and the member that occurs next ("next"). An additional item is included in this structure to enable the members to handle variable length items. The "len" will represent the amount of memory required by the data but will not include the overhead of the QITEM structure. As you can see, if each member maintains its QITEM structure, it will be possible to reach the items that precede and follow it by referencing the structure variables next or prev.

QMAIN is a header structure for the queue of items. This structure contains a pointer to the first QITEM (top) and a pointer to the last QITEM (bot). Additionally, the queue header will contain a counter (cnt) of the number of items contained in the list. Notice that the order of the structure variables "top" and "bot" is important. By aligning "bot" in the same position as "prev" in the QITEM structure and "top" with the "next" position, it is possible to make a complete "circle" through the items. Of course you don't want to accidentally treat the QMAIN as an element but there are times when you will intentionally treat the queue's head as a QITEM. (More on this later.)

Given these two structure definitions it is now possible to write code to handle doubly linked list. The most important part of this implementation is the queue functions that will do all the tedious work for you. You don't want the burden of maintaining the queue everywhere you use it, so Listing Two has the basic functions required.

At the very least, three of these functions are required, but it has been my experience that these three are enhanced by the addition of the other functions shown in the listing. The 3 primary functions are "qinit()", "qadd()", and "qdel()".

Before using a queue for the first time it is necessary to initialize the QITEM pointers, top and bot. The function qinit() will "point" the start and end pointers to the QMAIN itself and set the count of items to zero. It is important to initialize a queue once before using it. Using a non-initialized header in one of the subsequent functions will certainly lead to doom. Be sure to only initialize a queue when you know that it has not been used before. Initializing a queue that contains elements will result in your losing all the elements and the memory associated with those elements.

The second primary function is qadd(). A list is of little value if you cannot add new things to it. Notice that the parameters to this function include the QMAIN of the queue being referenced, a QITEM pointer to a member of the queue, the data's pointer and the length of data to be stored. You might think to yourself, "Why pass an element pointer when you are simply adding?" The answer is that this function actually "inserts" a new member. This functionality means that you could build a sorted list, if you first determine where the item should be placed. In actuality, the same code is required whether you are inserting in the middle of a queue or simply adding to the end of the queue. Therefore, the added feature only "costs" you a parameter value.

Remember how I stressed that the position of a QMAIN's top and bot pointers was important? Adding a new element demonstrates why. If you are adding a new element, you have to point to the previous element (the current last element) and

make it point to the new one. Likewise, you have to make the current element (which in the case of addition is the QMAIN) and make its previous pointer point to the new element. Finally, you have to make the new element point at the previous end and point the next at the QMAIN. By ensuring that the pointer in a QMAIN can be referenced like a QITEM, it is not necessary to add code to specifically check to see if you are addressing the QMAIN itself. All you have to remember is this: To add a new element at the end of a queue pass the address of the queue as the first two parameters.

Also notice how `qadd()` actually stores the data. First, the size of a QITEM is added to the length of the data as an overhead to maintain the doubly linked list. In this implementation, this means that the largest element that can be stored is 65525-bytes long (65535 minus 10, or the maximum value of an unsigned short minus the size of the QITEM structure). It has not been my experience that this size limitation causes any problems in programs.

The third required function is `qdel()`. As the name implies, this function is used to remove an element from the list. Removing an element from the list works almost in the reverse of the add code. First, point to the previous element. Set this pointer's next to the element following the one being deleted. The following element also has to be changed to point back to the prior element. Once these two element pointers have been changed, the member in question has been removed from the chain and the associated memory can be released.

Notice that the `qdel()` function returns a pointer to the element that followed the one deleted. This is most often the desired results since these functions are often reference in "loops". By returning the pointer to the next element you can easily keep track of where you are in the queue. Also, note that a special check is made to ensure that you do not accidentally try to remove the QMAIN from memory. The QMAIN is not an element and trying to release the memory associated with that pointer can be devastating.

Initialization, addition, and deletion are the primary functions required to make queuing work. In practice, however, the need for several other functions arise to enhance the useability of linked lists. Although there are a great number of functions that can be written to aide the use of queues (some of which I will cover in later articles), I think the ones that I have included constitute the ground level. A discussion of each follows.

Often it will be necessary to point at (and perhaps retrieve) a particular element in the list. To accomplish this, use the `qset()` function to return the QITEM pointer of the element you require. As an additional feature, and to help prevent errors, `qset()` expects a pointer to the variable that contains the offset position. This allows the function to alter the requested value if it falls outside the range of the queue. A small attempt is made to increase the speed of the locating process by determining whether the requested offset is in the first or second half of the queue.

Another positioning function that is useful is the opposite of `qset()`. This function, named `qpos()`, will return the relative position of an element in a queue. This function expects a QITEM pointer to an element and searches for it in the queue indicated by the other parameter. If the element is not part of that queue, or is the QMAIN, a value of zero is returned.

Updating existing elements is a common task for programs that use queues. Rather than making the programmer include code to remove an old element to add the revision, the function `qput()` can be included. Accepting the same parameters as

`qadd()`, the `qput()` function first determines whether the new data will fit into the same size location that is already used by the old element. If it will not, the old data is removed and the new data is added in its place. If it will, the data is simply moved into the old location.

Finally, when your program is finished using a queue it is necessary to release the memory associated with the elements. The `qfree()` function does this task by deleting each element that remains in a queue. Remember, if you don't release the memory, your program cannot reuse it. This is an important point to note, especially if the queue is a variable on the stack that will be lost after your function returns. Also, be careful not to simply re-init the queue using `qinit()`. This will definitely cause you to "lose" memory.

As mentioned, there are a number of other functions that could be added to this list to aide in the usefulness of queues. In the next couple of months I will probably include several others, but these are the key functions. Of course, a number of enhancements can be made to these functions to improve their performance both in speed and integrity. For instance, I also use "fixed-length" queues. These require similar functions (with "qf" as the prefix) that do not require you to pass lengths — except for `qfinit()` which stores the fixed length indicated in the QFMAIN structure. Additionally, other checks can be performed to ensure that elements from the "wrong" queue are not accidentally changed. But I will leave many of these types of enhancements to you the programmer.

Relying on queues and queue functions can be both wonderful and terrible. Wonderful because of their flexibility in manipulating all types of data easily, and terrible when one or more of the functions has a bug in it. As a precautionary measure, it is good to do a little Quality Assurance (QA) on the functions and I have provided such a program in Listing Three. In addition to being a test, this program also demonstrates how to use the queue functions that I have included.

The test program will test most of the features of the queue functions. A comment block has been provided in each area to explain the test that is being performed. Notice that in many area the entire queue is traversed either forwards or backwards. This is accomplished by making a pointer to store the current QITEM. Adding 1 to a QITEM pointer [i.e., `(ptr + 1)`] actually tells the compiler to add the "size" of a QITEM to the pointer. This is a quick method that can be used to point passed the overhead structure to the actual data that has been stored as an element.

Eight major tests are performed on the functions provided. Within each major test, several minor tests may also be done to check the reliability of the queue. In the event of a failure, the variable "error" will be assigned a unique number where it failed. If at the end of the tests, your program prints an error value other than zero, it either means that the test program or one of the queue functions has a bug. Use the error number to find the specific area that failed and then you will know what to begin checking. For instance, an error of 7 indicates that in test 3, the data in the queue does not match the expected value of count. This might mean that the `qadd()` function has a bug in storing the data properly.

Linked lists might look like a lot of work but once you have several well established functions, you will be surprised how many solutions to everyday programming problems they provide. They can offer more efficient memory use, which increases the useability of your programs. Take some time and do a few experiments. I hope you find queuing as useful as I do.

Listing One

```
typedef struct x {
    struct x    *prev;
    struct x    *next;
    unsigned short len;
} QITEM;

typedef struct {
    QITEM      *bot;
    QITEM      *top;
    unsigned long cnt;
} QMAIN;

void qinit(QMAIN *que);
QITEM * qadd(QMAIN *que, QITEM *member, void *data, unsigned short len);
QITEM * qdel(QMAIN *que, QITEM *member);
QITEM * qset(QMAIN *que, unsigned long *item);
unsigned long qpos(QMAIN *que, QITEM *ptr);
QITEM * qput(QMAIN *que, QITEM *ptr, void *data, unsigned short len);
void qfree(QMAIN *que);
```

Listing Two

```
#include "queue.h"
#include <stdlib.h>

/*-----*/
/* QINIT establishes a new Queue by pointing the top and bottom */
/* pointers to the head of the queue and sets count of items to */
/* zero. WARNING: if you do this function is passed an "active" */
/* queue then the memory and elements that it contains will be */
/* lost. */
/*-----*/

void qinit(QMAIN *que)
{
    que->top = (QITEM *)que; /* point top to MAIN */
    que->bot = (QITEM *)que; /* point bottom to MAIN */
    que->cnt = 0L;           /* No items in queue yet */
}

/*-----*/
/* QADD is used to add a new item to the queue in front of the */
/* QITEM pointer that was passed. To add an element to the end */
/* of a queue the pointer to the QMAIN should be passed as the */
/* member. Because of the overhead required to maintain the */
/* QITEM links the maximum element size that can be added is: */
/* (maximum unsigned short value - sizeof(QITEM)) or 65535 - 10 */
/*-----*/

QITEM * qadd(QMAIN *que, QITEM *member, void *data, unsigned short len)
{
    unsigned short reqsize;
    QITEM      *memaddr, *memprev;
    reqsize = len + sizeof(QITEM); /* adjust size */
    if ((memaddr = (QITEM *) malloc(reqsize)) == NULL) /* get memory */
        return((QITEM *)NULL); /* no memory */
    memprev = member->prev; /* point to item before member */
    memaddr->prev = memprev; /* make new item's previous pointer */
    memaddr->next = member; /* new item's next is the member */
}
```

```
memprev->next = memaddr; /* the previous item now points to new */
member->prev = memaddr; /* the next item points back to new */
memaddr->len = len; /* the length of the item stored */
movmem(data, (memaddr+1), len); /* store the data passed */
que->cnt++; /* count increments for this queue */
return(memaddr); /* return the QITEM pointer to new item */
}

/*-----*/
/* QDEL will remove the item pointed at by the passed pointer */
/* from the queue indicated. */
/*-----*/

QITEM * qdel(QMAIN *que, QITEM *member)
{
    QITEM *ptr;
    if (member == (QITEM *)que) /* first check to see if pointer is MAIN */
    {
        return(member); /* if so then return (nothing to do) */
        ptr = member->prev; /* point to the item before the element */
        ptr->next = member->next; /* it now points to the item after member */
        ptr = member->next; /* point at the item after the element */
        ptr->prev = member->prev; /* it now points back to the item before */
        que->cnt--; /* the queue has one less element */
        free(member); /* free the item */
        return(ptr); /* return the pointer to the next item */
    }

    /*-----*/
    /* QSET is used to return a pointer to the element at the */
    /* sequential position indicated by the value in item. If */
    /* item is greater than the number of items in the queue then */
    /* item is set to the count of the last item. */
    /* In order to speed the process of positioning (somewhat) */
    /* it is first determined if the item can be found faster by */
    /* starting from the last item in the queue or the first. */
    /*-----*/

    QITEM * qset(QMAIN *que, unsigned long *item)
    {
        unsigned long start;
        int inc;
        QITEM *ptr;

        if (*item > que->cnt) /* check to see if in this queue */
            *item = que->cnt; /* cannot be longer than queue */
        if (*item == 0) /* Check for a value of zero */
            return((QITEM *)que); /* return the pointer to MAIN */
        if (*item > (que->cnt >> 1)) { /* if item is in second half */
            inc = -1; /* look backwards */
            start = que->cnt; /* start from the end */
            ptr = que->bot; /* point at last item */
        } else { /* in the first half */
            inc = 1; /* look forward */
            start = 0; /* start from the beginning */
            ptr = que->top; /* point at first item */
        }
        while (ptr != NULL) {
            if (*item == start)
                return(ptr);
            ptr = ptr->next;
            start = start + inc;
        }
        return(ptr);
    }
}
```



```

start = 1; /* start at position one */
ptr = que->top; /* point to first item */
}
for ( ; *item != start; start += inc) /* look for item */
    ptr = (inc > 0) ? ptr->next : ptr->prev; /* forward of
backwards */
return(ptr); /* return the pointer to item */
}
/*-----*/
/* QPOS is used to determine the relative position of an item in */
/* the queue indicated. If the item is not found then zero is */
/* returned. The function starts at the MAIN queue position. */
/*-----*/
unsigned long qpos(QMAIN *que, QITEM *ptr)
{
    QITEM *srch;
    unsigned long item;
    srch = (QITEM *)que; /* point the header */
    for (item = 0L; item <= que->cnt; item++){ /* check entire queue */
        if (srch == ptr) return(item); /* if found, return
count */
        srch = srch->next; /* go to next item */
    }
    return(0L); /* does not occur in
queue */
}
/*-----*/
/* QPUT is used to re-store an element back into the queue at the */
/* same position that it held. If the element is the same size */
/* it is simply moved into the same spot. Otherwise, to old item */
/* is removed and the new item added in its place. */
/*-----*/
QITEM * qput(QMAIN *que, QITEM *ptr, void *data, unsigned short len)
{
    if (ptr->len == len){ /* if data fits into same slot */
        movmem(data, (ptr+1), len); /* move data into old space */
        return(ptr); /* and return this element */
    }
    ptr = qdel(que, ptr); /* remove the old pointer */
    return(qadd(que, ptr, data, len)); /* add the item back here */
}
/*-----*/
/* QFREE will free all the memory associated with the members of */
/* the queue indicated as the parameter. */
/*-----*/
void qfree(QMAIN *que)
{
    for(; que->cnt > 0;){
        qdel(que, que->top);
    }
}
/*-----*/

```

Listing Three

```

/*-----*/
/* This program will test the integrity of the queue */
/* functions. It will perform several test to ensure */
/* that each is working properly. */
/*-----*/
#include "queue.h"
#include <stdio.h>

typedef struct { /* element structure that is used to */
    unsigned short count; /* test the integrity of the queues */
    unsigned short len;
    char data[31];
} ELEMENT;

void main()
{
    QMAIN queue;
    ELEMENT item;
    QITEM *ptr;
    unsigned long pos;
    unsigned count;
    unsigned error;

    qinit(&queue); /* initialize the queue */

    /*-----*/
    /* First check to ensure that the top and bottom pointers */
    /* point to the queue's address when after initialization. */
    /*-----*/
    printf("Begin Test 1\n");
    error = 0; /* assume no errors */
    if (queue.top != (QITEM *)&queue) /* is top incorrect */
        error = 1; /* set the error condition */
    if (!error && queue.bot != (QITEM *)&queue) /* is end incorrect */
        error = 2; /* set error condition */
    if (!error && queue.cnt != 0L) /* length is not correct */
        error = 3; /* set error condition */

    /*-----*/
    /* Next we will add to the queue 300 elements of */
    /* varying lengths and data. */
    /*-----*/
    if (!error){ /* if no errors yet */
        printf("Begin Test 2\n");
        for(count = 1; count <= 300; count++){
            item.count = count; /* set the item count */
            item.len = count % 30; /* set variable length */
            if (item.len == 0) /* on zero set to 30 */
                item.len = 30; /* reset */
            memset(item.data, 35+item.len, item.len); /* random data */
            ptr = qadd(&queue, (QITEM *)&item, (unsigned
short) (item.len+4));
            if (ptr == NULL){ /* if failed to add */
                error = 4; /* indicate error */
                break; /* end the test */
            }
            if (count != queue.cnt){ /* check counts */
                error = 5; /* indicate error */
                break;
            }
        }
    }

    /*-----*/
    /* Now verify that the entire queue can be traversed in both */
    /* directions and verify that the proper data was stored. */
    /*-----*/
    if (!error){ /* as long as no error yet */
        printf("Begin Test 3\n");
        count = 0; /* assume no value yet */
        for (ptr = queue.top; ptr != (QITEM *)&queue; ptr = ptr->next){
            if (count == 300){ /* failed to find end */
                error = 6; /* set error */
                break;
            }
            count++; /* increment the count */
            movmem((ptr-1), &item, sizeof(item)); /* move largest size */
            if (item.count != count){ /* verify data */
                error = 7; /* error */
                break; /* stop */
            }
            if (item.len != count % 30){ /* check proper size */
                if (! (item.len == 30 && (count % 30) == 0)){ /* except */
                    error = 8; /* set error */
                    break; /* stop */
                }
            }
            if (item.len != ptr->len-4){ /* check length stored */
                error = 9; /* set error */
                break; /* stop */
            }
        }
    }
}

```



```

    if (item.data[0] != 35+item.len){
        error = 10; /* wrong data */
        break;
    }
    if (!error && count != 300){ /* not enough items */
        error = 11; /* indicate error */
    }
}

if (!error){ /* as long as no error yet */
    count = 0; /* assume no value yet */
    for (ptr = queue.bot; ptr != (QITEM *)queue; ptr = ptr->prev){
        if (count == 300){ /* failed to find end */
            error = 12; /* set error */
            break;
        }
        count++; /* increment the count */
    }
    if (count != 300) /* not enough items */
        error = 13; /* indicate error */
}

/* Now test the positioning within the queue using the */
/* position by offset and by pointer functions. */

if (!error){ /* if no errors yet */
    printf("Begin Test 4\n");
    pos = 100; /* position in first half */
    ptr = qset(&queue, &pos); /* get the pointer */
    if (pos != 100) /* proper place */
        error = 14; /* incorrect value */
}

if (!error){
    if (qpos(&queue, ptr) != pos) /* is this the same place */
        error = 15; /* nope */
}

if (!error){
    if (((ELEMENT *) (ptr+1))>count != pos) /* check data */
        error = 16; /* nope */
}

if (!error){ /* if no errors yet */
    pos = 200; /* position in second half */
    ptr = qset(&queue, &pos); /* get the pointer */
    if (pos != 200) /* proper place */
        error = 17; /* incorrect value */
}

if (!error){
    if (qpos(&queue, ptr) != pos) /* is this the same place */
        error = 18; /* nope */
}

if (!error){
    if (((ELEMENT *) (ptr+1))>count != pos) /* check data */
        error = 19; /* nope */
}

if (!error){
    pos = 500; /* set invalid size */
    if (qset(&queue, &pos) != queue.bot) /* does it return end */
        error = 20;
}

if (!error){
    if (pos != queue.cnt) /* did it reset pos */
        error = 21;
}

if (!error){
    pos = 0; /* set MAIN */
    if (qset(&queue, &pos) != (QITEM *) queue)
        error = 22;
}

if (!error){
    ptr = NULL; /* set a bad pointer */
    if (qpos(&queue, ptr) != 0) /* should not find this */
        error = 23;
}

/* Now test the deletes code by removing all the odd numbers elements. */

if (!error){
    printf("Begin Test 5\n");
    for (ptr = queue.top; ptr != (QITEM *)queue; ptr = ptr->next){
        if (queue.cnt < 150){ /* too many */
            error = 24;
            break;
        }
        ptr = qdel(&queue, ptr); /* delete returns next pointer */
    }
    if (!error && queue.cnt != 150)
        error = 25;
}

if (!error){ /* check for only evens */
    for (ptr = queue.top; ptr != (QITEM *)queue; ptr = ptr->next){
        if (((ELEMENT *) (ptr+1))>count % 2 != 0){ /* oops */
            error = 26;
            break;
        }
    }
}
}

```

```

/* Now check the code that puts data back into the queue. */
/* This time use a fixed size for all items so that some */
/* should fit into the existing space while others have to */
/* be deleted and added again. */

if (!error){
    printf("Begin Test 6\n");
    for (ptr = queue.top; ptr != (QITEM *)queue; ptr = ptr->next){
        movmem((ptr+1), &item, ptr->len); /* get the data */
        memset(item.data, ' ', 15); /* reset data */
        item.len = 15; /* reset length */
        ptr = qput(&queue, ptr, &item, (unsigned short) (item.len+4));
        if (ptr == NULL){ /* if out of mem */
            error = 27;
            break;
        }
    }
}

/* After the puts, check to insure that the integrity of the linked */
/* list has been maintained by traversing the queue in both */
/* directions. */

if (!error){
    printf("Begin Test 7\n");
    count = 0;
    for (ptr = queue.top; ptr != (QITEM *)queue; ptr = ptr->next){
        if (count == 150){ /* too many */
            error = 28;
            break;
        }
        count++;
    }
    if (count != 150 && queue.cnt != 150) /* check counts */
        error = 29;
}

if (!error){
    count = 0;
    for (ptr = queue.bot; ptr != (QITEM *)queue; ptr = ptr->prev){
        if (count == 150){ /* too many */
            error = 30;
            break;
        }
        count++;
    }
    if (count != 150) /* check count */
        error = 31;
}

/* As a final test, the free code is called to release all the */
/* elements from the queue. Afterwards the top and bottom */
/* pointers should point to the queue's address. */

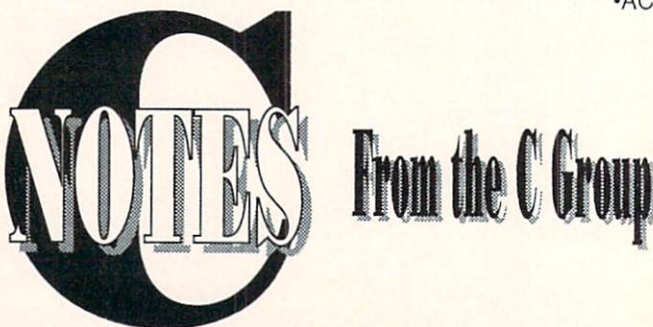
if (error == 0 || error > 3){ /* if queue init'd */
    printf("Begin Test 8\n");
    qfree(&queue); /* empty */
    if (!error){
        if (queue.top != (QITEM *)queue)
            error = 32;
    }
    if (!error){
        if (queue.bot != (QITEM *)queue)
            error = 33;
    }
}

/* As a verification to how successful these tests have been, print the */
/* error code's value. If error has a value other than zero then there */
/* is either an error in the queuing functions or in this test code. */
/* Use the number printed to determine where the failure occurred. */

printf("Error = %d\n", error);
}

```

•AC•





Display à la mode

by Ernest P. Viveiros, Jr.

ALONG WITH THE NEW WORKBENCH 2.0 software comes new display modes. Unlike the two previous display modes for the Amiga (normal and Interlace), these enhancements are more easily accessible by the user. They also add a great deal more flexibility and choice for the Workbench display.

Selecting a Workbench display mode is accomplished simply by using the WBScreen editor. To do this, open the Preferences drawer, and double-click the WBScreen icon. Welcome to the WBScreen editor! The available display modes, depending on the system hardware and monitors, are shown in a scroll gadget. Just point and click on the mode of your choice. Each display mode has certain additive properties. A display box in the WBScreen window displays the properties of the selected display mode. Possible display properties include:

Interlaced	The display mode supports an interlaced screen.
ECS	Uses the Enhanced Chip Set.
PAL	PAL display mode available. (Only on NTSC Amiga with ECS.)
NTSC	NTSC display mode available. (Only on PAL Amiga with ECS.)
Supports Genlock	Supports the use of genlocking hardware.
Draggable	Supports a draggable Workbench screen.
Panelled	The selected display mode is made up of several panels.

Also displayed in the WBScreen window is information about the screen sizes of the currently selected display mode. Information displayed includes:

Visible Size	The size of the text over-scan area.
Min Size	The smallest size that the display mode allows.
Max Size	The largest size that the display mode allows.
Max Colors	The maximum number of colors that can be displayed on a screen.

This is where it gets good! In the WBScreen window, next to the screen size information, there are two text gadgets labeled "Width" and "Height". Here you can set the size of the display mode. Your only limits are Min Size, Max Size and the available CHIP RAM.

Hires and Super Hires are among the Workbench 2.0 display types.

After setting the Width and Height of the display mode, you can set the number of colors that can be displayed. Using the Color slider gadget, you can increase or decrease the number of colors. Again, this is limited by Max Colors and available CHIP RAM.

Finally, if you create a display mode wider than the monitor's display area, you can select the AutoScroll feature. AutoScroll scrolls the screen automatically when the mouse reaches the edge of the visible screen. It's fast and smooth.

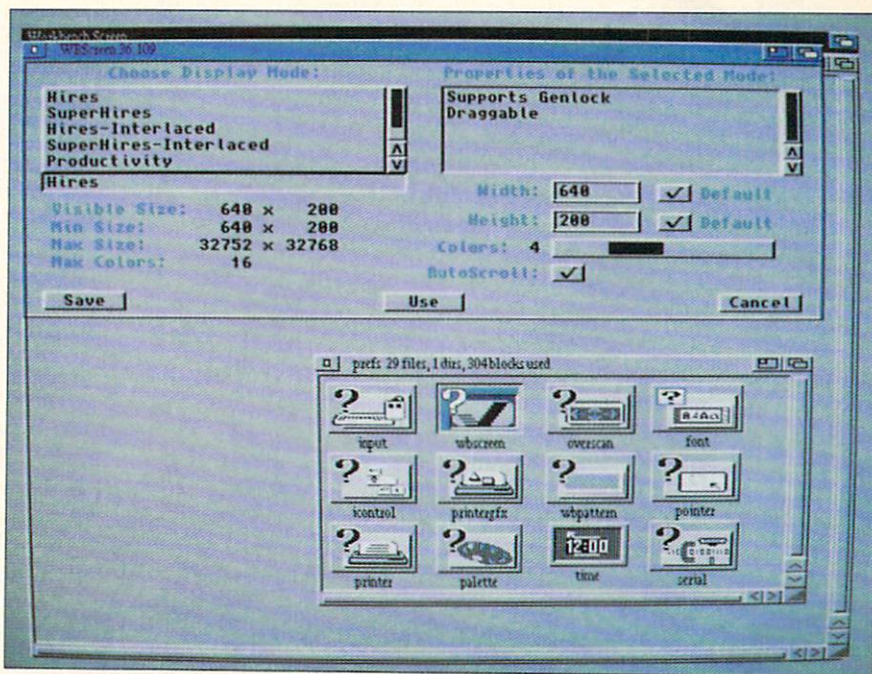
Good in theory, and in practice

Using an Amiga 3000/25 (with only 1MB CHIP RAM) I have routinely created and worked with Hires-Interlaced displays which would be the size of a 4-page display (approximately a 20" x 25" display...it's hard to measure a scrolling display).

The Soapbox

These new display modes in addition to the oversize display bring new life (and powerful life) to the Amiga. However, I really can't do anything spectacular with it right now. Much of the current 1.3 software runs buggy with plain vanilla Workbench 2.0, never mind with oversize displays. If only more Amiga developers would create applications that take advantage of the Amiga's unique Workbench display modes. The Amiga is ready.

•AC•



PIPE DREAM

by Miguel Mulet

Pipe Dream is Lucasfilm's latest entry in the arcade game market. In it, you play a plumber trying to control the excess "floozy" being produced by Acme Chemicals ("floozy" is a new brand of sewer cleaner). To accomplish this, you must build a pipeline long enough to contain the "floozy" coming from the plant. You score points for every piece of pipeline through which the "floozy" flows, and lose points for any piece of pipeline through which it doesn't flow. What makes the pipeline long enough? It depends on the level you're on—the higher the level, the longer the pipeline required. The upper right corner of the screen shows what level you're on (L:) and the distance in pieces the "floozy" must travel (D:).

To build a pipeline, you take pieces from the parts bin on the left side of the screen. You really don't get to "select" each piece—you must use whatever is the next available piece from the bin. Pieces are placed on the 7 X 10-piece playing field by moving the cursor (using the joystick, mouse, or keyboard) over the desired location, and then pushing the fire button. You cannot move a piece once it's placed, but you can put another piece on top of it. Replacing a piece takes a little longer, and also decreases your score by 50 points.

The pieces of pipe come in assorted shapes and sizes. There are four right-angle pieces (which turn the "floozy" 90 degrees), horizontal, and vertical pieces, as well as cross-shaped pieces which allow



BATTLE SQUADRON

by Tony Preston

This game was first shown at the World of Commodore show in Valley Forge, PA. It looked something like Hybris, one of my favorite "shoot-em-up" arcade style games. After watching my sons play for a while, I realized that it is a lot more. I talked to Paul Lombardi, the gentleman at the booth and President of InnerPrise, and found out that it was created by the authors of Hybris. After talking for awhile, I arranged for a review copy. InnerPrise Software has a real winner here. This game has all the elements needed to satisfy the most ardent arcade adventurer!

Battle Squadron starts out with a short series of black and white pictures and a story of the attack of the Barrax Empire. It seems that the Earth defense forces must

rescue two comrades that were captured returning from a secret mission to Urania, the Barrax Empire home planet. The information they have will make the difference between success and failure of future counteroffensive actions! Commander Mayers and Bergin only managed a partial transmission of warning before they were captured. Your mission is to eliminate all Barraxian life forms from the planet's surface and subterranean levels to rescue your comrades! If you are successful, rescuing your comrades you help to defeat the Barrax Empire. The pictures and story at the beginning set the scene and can be bypassed with a click of the fire button.

Battle Squadron is a one or two player game, at the start you can select your

you to form "loops" in the pipeline. If you can get the "flood" to cross itself in the pipeline, additional points are awarded. Any of the pieces above (except the cross) may appear as a "one-way valve"—meaning that the "flood" will stop if the piece is placed in the opposite direction of the pipeline flow.

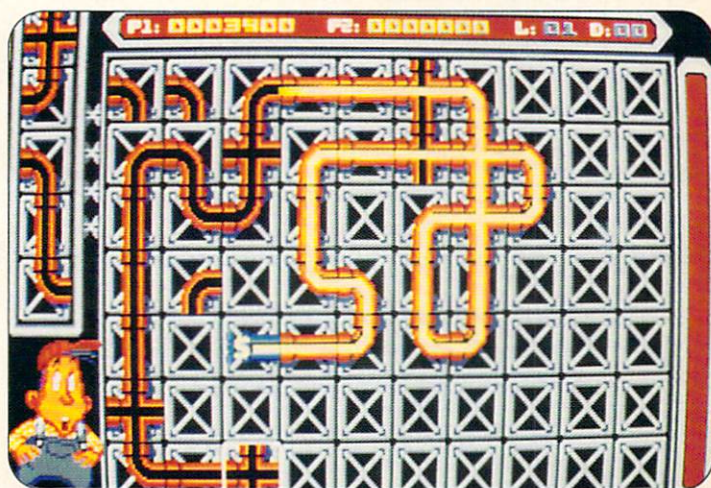
Some levels include "bonus" pieces. If you can get the "flood" to flow through them, more points are awarded. Of particular interest are the reservoir pieces, which slow the "flood" temporarily (giving you a chance to catch your breath!). As if life as a plumber isn't hard enough, there are often obstacles (such as fire hydrants, broken sidewalks, etc.) which prevent you from running the pipeline through that space. Lastly, if you place pieces on the "board" through which the "flood" doesn't travel, you lose points.

Pipe Dream is provided on one non-copy-protected disk. The manual encourages you to make a backup copy and use that for game play. To start the game, insert the Pipe Dream disk at the Workbench prompt. Although the disk is copyable, you do need the code wheel to begin. Once the title screen is shown, the game displays a picture and the name of a color. Using the code wheel, you dial in the correct picture

and read the code which appears under the correct color. After the correct code is entered (you get three tries), the main menu is revealed, allowing you to customize the game.

There are several different playing modes. In the trainer mode, "flood" travels slowly, giving you time to study the level and place the pieces. "Flood" speeds up in the basic mode, but gameplay is otherwise the same. The expert mode allows a player to choose pieces from two different pipe bins. You select which bin the next part will come from by pressing the fire button and moving the joystick in the direction of the bin you want the next piece to come from. The advantage of this is two-fold: you get

double the number of pieces, and 100 bonus points for getting the "flood" to flow through pieces selected from alternating bins (and you thought plumbing was easy!). The two-player mode allows the participants to place pieces both competitively and cooperatively—but each player



Pipe Dream by LucasFilms Games



options such as joystick or mouse, one or two players, etc. You can turn the music and sound effects off individually, but they add a lot to the excitement! When I first saw this game it was playing on an A500 hooked to a stereo, the sounds were incredible! This game really uses the Amiga stereo sound to its fullest. There is over 30 minutes of true stereo, digitized sounds, and music in a special format that only takes up about 38 K on the disk. Because of the special compression techniques, InnerPrise is considering marketing a music editor using this format and plans to license the technology to other developers.

Similar to Hybris, is an optional setup screen where you can adjust specific game play features and difficulty. I like to setup the game with slower enemies and the maximum number of lives. That way I survive a lot longer!

My favorite weapon is the Anti-Matter Particle Beams (Blue weapon) when upgraded, it will fire both forwards and backwards! There are 25 different weapon configurations. As you fight along your assigned mission, you will run into a ship that

is heavily armored and armed. If you destroy the ship, it will leave a floating pod which when scooped up will either switch or upgrade the current weapon. The color of the pod is what determines which weapon it will upgrade. You have one additional weapon that is very useful in tight situations, the Nova Smart Bomb. Nova Smart Bombs are packed with terrific destructive power, eliminating all enemies near you. You start the game with three Nova Smart Bombs, but will find floating capsules marked with an "M" that you can use to increase your supply.

The instruction manual for the game is fairly simple and straightforward. It gives all the essential information on weapons and options. It even includes your orders from Headquarters!

Once things are setup and you start, the action is fantastic. The screen scrolls smoothly in all four directions as you fly your fighter. The basic idea of the game is

Battle Squadron from InnerPrise Software



simple, shoot everything in sight! There are ground gun and missile emplacements to watch out for in addition to the many different attacking aircraft.

Battle Squadron has a nice feature, with two players during level one, you can get killed and restart joining your comrade again and again! If you get killed while still

scores points only when flooz flows through pieces he or she has actually placed. Any combination of joystick/keyboard, mouse/joystick, etc. can be used in the two-player mode. Once you select the number of players and what input devices they'll be using, you can then specify the level at which you'd like to start by typing in the password for that level (you get the password by successfully completing the previous level).

Pipe Dream is a well-thought-out action-strategy game. The player must fight against time in order to build a continuous pipeline with the pieces available. If a piece has no immediate use, the player must think ahead as to where the pipeline may go. Flexibility is also important, as players may find that they have to reroute the pipeline in order to meet a specific goal of the level they're on. The two-player game is particularly interesting, as each player scores points for the amount of "flooz" that flows through his/her pieces. If players don't cooperate, the "flooz" won't flow long enough, and both players will lose. On the other hand, once the "flooz" has flowed through the required number of pieces on a level, both players are at each other's throats trying to get the "flooz" through their equipment. An interesting arrangement to say the least!

Pipe Dream has some other nice features. For instance, you can restart at almost any level desired. After every four levels, a password is displayed on the screen. Enter that password the next time you play and the game bypasses earlier levels, returning you to the level at which you left off. You are encouraged to make a backup copy to play the game; this is important because the game does store the high scores to disk. There are also two ways to pause the game. Once you have completed a level, the computer requires

you press the fire button to go to the next level (this allows you to regain your composure after a particularly grueling level). Also, if the phone should ring while you're playing, hitting the 'P' key will pause the game in mid-stride.

Pipe Dream also has some drawbacks. Game graphics and sound are adequate but not spectacular, and the theme music that plays at the beginning of each level tends to grate on your nerves after a while. The documentation is also confusing at first. The main manual is written for the Macintosh, and the screen dumps are from the Mac as well. The code wheel used for copy protection has some rather generic symbols, while the Amiga uses graphics which differ from both the code wheel and the manual. Thus, it took a little bit of time to decipher the codes when things didn't look too much alike. The "Amiga specific" reference card also makes reference to "making a folder"—another carryover from the Mac. It would be nice to see at least a reference card which was truly written for the Amiga.

Pipe Dream is not the type of game that you can't put down, but the better you do, the more you want to play. Game play is both interesting and challenging. The two-player mode is great fun. I like Pipe Dream, although it's not the type of game that blows your socks off. Take a pipewrench to your local dealer and give Pipe Dream a try.

•AC•

PIPE DREAM
Lucasfilm Games
P.O. Box 10307
San Rafael, CA 94912
(415) 662-1902
Price: \$39.95
Inquiry #206



on level one, you don't have to wait for your partner to die before playing again! A quick click or two of the fire button puts you back in the action! Once you leave level one and proceed into the subterranean levels (level two and onward), dead is dead, you have to wait till your partner dies before joining in again.

You start out on the planet's surface fighting your way to the entrance to the next level. The entrance to the second level is well marked and takes you inside the planet where you come up against a wide variety of creatures. These creatures are all out to end your mission in a ball of flames! At the end of the second level you return to the upper surface for the third level. Again fight your way to the fourth level entrance and go down to face more of the planet's creatures. You alternate above and below the surface with things getting more and more difficult at each level. Eventually, you reach the end of what seems like an impossible mission for one last battle. If you succeed, you are declared a hero and have rescued your comrades. It isn't easy at all to get thru the different levels, I died many a time. I did finally make it thru to the very end (I had the help of a magic word that makes you invincible). Without the help, I don't think my reflexes are fast enough to defeat the Barraxian Empire! Maybe you can succeed where I failed!

I had a lot of fun with this game of fast action, super graphics, and superior sound. From an enjoyment standpoint this game is the best I've seen. The Graphic images for the ground emplacements have a 3-D effect that has to be seen. The pyramid shaped gun and

missile emplacements not only have a feeling of depth, but down below are more targets for you to destroy!

The Graphics uses 32 colors, with vertical and horizontal scrolling that is the smoothest of any game I have played. With over 30 different enemies to worry about, you rarely have time to rest. In some situations, you may have as many as 70 objects on the screen at the same time! The game has several levels of different scenery with each containing more deadly enemies to attack. You actually have a choice at the end of level one, you can bypass the passage into the next level and continue on to see more of the first level. Once you leave the first level and then get killed, your game is over. The game allows play with both the mouse and a joystick(my preference).

The game is great, well worth the suggested retail price of \$39.95. Battle Squadron runs on any Amiga and only requires 512K. An update is planned with additional sound capabilities that will only be available if you have 1 meg or more of memory. The game comes on a copy protected disk.

If you enjoy a fast paced shoot-em-up arcade type game Battle Squadron is the game for you!

•AC•

BATTLE SQUADRON
InnerPulse Software, Inc.
128 Cockeysville Road
Hunt Valley, MD 21030
(301) 785-2266
Price \$39.95
Inquiry #205

Listing One

LISTING #1, Drawing circles using convergence, AmigaBASIC

```

REM *****
REM *                               Program to test for convergence
REM *                               by Paul Castonguay
REM *                               December 13, 1988
REM *****

```

```

DEF FNx(x) = INT((x-xmin)+dx/2)/dx
DEF FNY(y) = 195 - INT((y-ymin)+dy/2)/dy
xmin = -2
xmax = 2
ymin = -1.4
ymax = 1.4
dx = (xmax-xmin)/631
dy = (ymax-ymin)/195

```

```

SCREEN 1, 640, 200, 4, 2
WINDOW 2, , (0,0)-(631,186), 0, 1

```

```

Crunch = 500
M = 4
CLS

```

```

LOCATE 10, 23
PRINT "... Pattern is being generated ..."

```

```

FOR j=ymin TO ymax+dy/2 STEP dy
  FOR i = xmin TO xmax+dx/2 STEP dx
    GOSUB Calculate
    GOSUB Select.Color
    PSET (FNx(i),FNY(j))
  NEXT i
NEXT j

```

```

WHILE INKEY$ <> CHR$(138)
WEND

```

```

WINDOW CLOSE 2
SCREEN CLOSE 1

```

```

END

```

```

Calculate:

```

```

  r = i*i + j*j
  k = 0
  WHILE r<=M AND k<Crunch
    r = r*r
    k = k+1
  WEND
  RETURN

```

```

Select.Color:

```

```

  IF k = Crunch THEN
    COLOR 2
  ELSEIF k>=11 THEN
    COLOR 1
  ELSEIF k>=5 THEN
    COLOR 12
  ELSEIF k=4 THEN
    COLOR 11
  ELSEIF k=3 THEN
    COLOR 10
  ELSEIF k=2 THEN
    COLOR 9
  ELSEIF k=1 THEN
    COLOR 8
  ELSE
    COLOR 0
  END IF
  RETURN

```

Listing Two

LISTING #2, Calculate subroutine for Mandelbrot set, AmigaBASIC

```

Calculate:

```

```

  x = 0
  y = 0
  k = 0
  r = 0
  WHILE r<=M AND k<Crunch
    xk = x*x - y*y + i
    y = 2*x*y + j
    x = xk
    k = k+1
    r = x*x + y*y
  WEND
  RETURN

```

Listing Three

LISTING #3, Modified Mandelrot subroutine, AmigaBASIC

```

Calculate:

```

```

  x = 0
  y = 0
  k = 0
  xk = 0
  r = 0
  WHILE r<=M AND k<Crunch
    xk = x*x*x - 3*x*y*y + x*x - y*y + i
    y = 3*x*x*y - y*y*y + 2*x*y + j
    x = xk
    k = k+1
    r = x*x + y*y
  WEND
  RETURN

```

Listing Four

LISTING #4, Listing to draw figure #2

```

REM *****
REM *                               Sixteen Color Fractal
REM *                               by Paul Castonguay
REM *****

```

```

DEF FNx(x) = INT((x-xmin)+dx/2)/dx
DEF FNY(y) = (Vertical.Pixels - 5) - INT((y-ymin)+dy/2)/dy

```

```

xmin = -.06
xmax = .04
ymin = 1.1345
ymax = 1.2045

```

```

Vertical.Pixels = 400

```

```

dx = (xmax-xmin)/631

```

```

IF Vertical.Pixels = 200 THEN

```

```

  dy = (ymax-ymin)/195
  SCREEN 1, 640, 200, 4, 2
  WINDOW 2, , (0,0)-(631,186), 0, 1
  ELSEIF Vertical.Pixels = 400 THEN
    dy = (ymax-ymin)/395
    SCREEN 1, 640, 400, 4, 4
    WINDOW 2, , (0,0)-(631,386), 0, 1
  ELSE

```

```

    PRINT "Set Vertical.Pixels to 200 or 400"
    GOTO Done
  END IF

```

```

GOSUB Choose.Color.Numbers
GOSUB Use.New.Colors

```



```

Crunch = 800
M = 4
CLS

LOCATE 10, 23
PRINT "... Fractal is being generated ..."

FOR j=ymin TO ymax+dy/2 STEP dy
  FOR i = xmin TO xmax+dx/2 STEP dx
    GOSUB Calculate
    GOSUB Select.Color
    PSET (FNx(i),FNy(j))
    IF INKEY$ = CHR$(138) THEN GOTO Done
  NEXT i
NEXT j

REM Wait for operator to press function key [F-10]
WHILE INKEY$ <> CHR$(138)
WEND

Done:
WINDOW CLOSE 2
SCREEN CLOSE 1

END

Calculate:
x = 0
y = 0
k = 0
xk = 0
yk = 0
r = 0
WHILE r<=M AND k<Crunch
  xk = x*x*x - 3*x*y*y + x*x - y*y + 1
  yk = 3*x*x*y - y*y*y + 2*x*y + j
  x = xk
  y = yk
  k = k+1
  r = x*x + y*y
WEND
RETURN

Select.Color:
IF k = Crunch THEN
  COLOR 0
ELSEIF k>=100 THEN
  COLOR 1
ELSEIF k>=31 THEN
  COLOR 2
ELSEIF k>=26 THEN
  COLOR 3
ELSEIF k>=21 THEN
  COLOR 4
ELSEIF k>=18 THEN
  COLOR 5
ELSEIF k>=16 THEN
  COLOR 6
ELSEIF k>=13 THEN
  COLOR 7
ELSEIF k>=11 THEN
  COLOR 8
ELSEIF k=10 THEN
  COLOR 9
ELSEIF k=9 THEN
  COLOR 10
ELSEIF k=8 THEN
  COLOR 11
ELSEIF k=7 THEN
  COLOR 12
ELSEIF k=6 THEN
  COLOR 13
ELSEIF k=5 THEN
  COLOR 14
ELSEIF k=4 THEN
  COLOR 15
ELSE

```

```

COLOR 0
END IF
RETURN

```

Choose.Color.Numbers:

```

LOCATE 10,20
PRINT "... Please wait while I adjust colors ..."
Reg.0.Red = 0
Reg.0.Green = 0
Reg.0.Blue = 0
Reg.1.Red = 15
Reg.1.Green = 15
Reg.1.Blue = 15
Reg.2.Red = 15
Reg.2.Green = 15
Reg.2.Blue = 0
Reg.3.Red = 11
Reg.3.Green = 15
Reg.3.Blue = 0
Reg.4.Red = 11
Reg.4.Green = 13
Reg.4.Blue = 6
Reg.5.Red = 12
Reg.5.Green = 11
Reg.5.Blue = 3
Reg.6.Red = 12
Reg.6.Green = 8
Reg.6.Blue = 0
Reg.7.Red = 13
Reg.7.Green = 6
Reg.7.Blue = 0
Reg.8.Red = 14
Reg.8.Green = 4
Reg.8.Blue = 0
Reg.9.Red = 15
Reg.9.Green = 0
Reg.9.Blue = 0
Reg.10.Red = 5
Reg.10.Green = 9
Reg.10.Blue = 13
Reg.11.Red = 2
Reg.11.Green = 9
Reg.11.Blue = 12
Reg.12.Red = 0
Reg.12.Green = 9
Reg.12.Blue = 11
Reg.13.Red = 3
Reg.13.Green = 9
Reg.13.Blue = 10
Reg.14.Red = 4
Reg.14.Green = 9
Reg.14.Blue = 9
Reg.15.Red = 0
Reg.15.Green = 9
Reg.15.Blue = 9
RETURN

```

Use.New.Colors:

```

PALETTE 0, Reg.0.Red/16, Reg.0.Green/16, Reg.0.Blue/16
PALETTE 1, Reg.1.Red/16, Reg.1.Green/16, Reg.1.Blue/16
PALETTE 2, Reg.2.Red/16, Reg.2.Green/16, Reg.2.Blue/16
PALETTE 3, Reg.3.Red/16, Reg.3.Green/16, Reg.3.Blue/16
PALETTE 4, Reg.4.Red/16, Reg.4.Green/16, Reg.4.Blue/16
PALETTE 5, Reg.5.Red/16, Reg.5.Green/16, Reg.5.Blue/16
PALETTE 6, Reg.6.Red/16, Reg.6.Green/16, Reg.6.Blue/16
PALETTE 7, Reg.7.Red/16, Reg.7.Green/16, Reg.7.Blue/16
PALETTE 8, Reg.8.Red/16, Reg.8.Green/16, Reg.8.Blue/16
PALETTE 9, Reg.9.Red/16, Reg.9.Green/16, Reg.9.Blue/16
PALETTE 10, Reg.10.Red/16, Reg.10.Green/16, Reg.10.Blue/16
PALETTE 11, Reg.11.Red/16, Reg.11.Green/16, Reg.11.Blue/16
PALETTE 12, Reg.12.Red/16, Reg.12.Green/16, Reg.12.Blue/16
PALETTE 13, Reg.13.Red/16, Reg.13.Green/16, Reg.13.Blue/16
PALETTE 14, Reg.14.Red/16, Reg.14.Green/16, Reg.14.Blue/16
PALETTE 15, Reg.15.Red/16, Reg.15.Green/16, Reg.15.Blue/16
RETURN

```

•AC•

The Fred Fish Collection

Due to the increasing size of the Fred Fish Collection, only the latest disks are represented here. For a complete list of all AC, AMICUS, and Fred Fish Disks, cataloged and cross-referenced for your convenience, please consult the current AC's Guide To The Commodore Amiga available at your local Amazing Dealer.

Fred Fish Disk 320

AmigaTrekA continuation of Mike's Amiga Trek stories, which are parodies of the Star Trek series, with an Amiga flavor. Earlier stories are on disk 278. Author: Mike Smithwick

AmiOmega

Amiga port of the Omega game. Omega is similar to hack or rogue, but is much more complex. There is a city, several towns, a wilderness, lots of dungeons, a multitude of monsters, lots of spells, magic items, etc. There are several quests to complete. All in all, it is an excellent game. Requires 1Mb or more of memory. Amiga version 1.0, binary only. Author: Laurence Brothers, Amiga port by Rick Golembiewski

Fred Fish Disk 321

DezHexBin An intuition based programmers tool to convert integers between decimal, hexadecimal, and binary. Very small. Version 1.1, includes source in assembly code. Author: Michael Davidson

IconJ

IconJ significantly enhances the IconK program, and is 100% compatible. It allows scripts to be executed by double-clicking the script's icon. Abilities include joining the script with the icon file itself, or calling it from any directory or disk, executing either AmigaDOS or AmigaDOS scripts, outputting to any file or device, running interactive scripts and scripts that contain conditions, and creating relative console windows. Includes a utility called Atsui which attaches or detaches a script to/from an icon file. Version 1.0, includes source in JForth. By: Rich Franzen

Its

An iterated Function System viewer which graphically displays iterated function systems and allows the user to interactively create the affine functions that define such systems. An IFS can represent complex pictures very compactly. Simple IFSs can describe an infinite number of different and interesting fractal displays. Includes a number of displays that the author and others have discovered. Version 1.5, includes source in C. Author: Glen Fuller

Planets

Some routines ported to the Amiga by Bob Leivan, that compute the location of the planets (as viewed from a specific point on the earth) and the phase of the moon, for an arbitrary date and time. Includes source. Author: Keith Brandt VIII, Jim Cobb, F. T. Mendenhall, Alan Paeth, Petr Laurianin, Bob Leivan

Turtle

A shared library of "turtle" functions for drawing in a RastPort. Includes source in assembly and C. Author: Thomas Albers

UnixDir

A program which intercepts calls to dos.library to add the UNIX style "ls" and "cd" syntax for current and parent directories, respectively, to file and path names. I.E., you can refer to files in the current directory as "foo" and files in the parent directory as "foo.", or any combination of the two. Includes source in assembly. Author: Murray Bennett and Mark Cyster

Whereis

Another "find-that-file" utility. Whereis searches on your (hard-)disk for a file (name) and displays the path to that file. Some features are case independent search, wildcards, interactive mode (cd implemented), can display size and date of files, always abortable, can archive filenames for "ZOO" (like trams/record). And no recursive procedures. Includes source in C. Version 1.18 (2-15-90). Author: Roland Bless

Fred Fish Disk 322

Gwin This is version 1.0 of GWIN. GWIN or Graphics Window is an integrated collection of graphics routines callable from C. These routines make it easy to create sophisticated graphics programs in the C environment. One-line calls give you a custom screen (ten types available), menu items, requestors, text, circles, polygons, etc. GWIN is a two-dimensional floating point graphics system with conversion between world and screen coordinates. GWIN includes built-in clipping that may be turned off for speed. Use of color and XOR operations are greatly simplified. Many examples of the use of GWIN are included in an examples directory. Examples include line bar graph program, geographic mapping program, SPICE 2G.6 graphics post-processor, and others. Extensive documentation is included. Author: Howard G. Anderson

Fred Fish Disk 323

ColorTools Three tools that manipulate the colors of your screen.

Binary only. Author: Dieter Bruns

CZed

A complete midi package for use with all Casio CZ synthesizers. Contains a full fledged sound editor, a split simulator for CZ-101/1000/230S, a bank loader and a memory dump for CZ-1. This is a formerly commercial package now released as shareware.

Binary only. Author: Oliver Wagner

LinkSound

Two examples of functions that you can link with your own code to produce a short musical "beep" or a sound that is similar to striking a drum. Includes source.

Author: Dieter Bruns

Show

A very versatile program to display IFF ILBM files. Features realtime unpacking scroll, smart analysis of any IFF file, total control over display modes, simple slideshow processing, pattern matching, and a dozen other options. Only 9K. Version 2.0, binary only.

Author: Sebastian Vigna

Fred Fish Disk 324

This Fred Fish Disk is offered as an abridged disk until Fred can create replacement disk. One program has been removed from this disk due to copyright problems.

ANSIED

Demo version of an ANSI screen file editor. It allows you to easily create and modify a screen of ANSI-style text/graphics on the Amiga. The standard ANSI color set (red, green, yellow, blue, magenta, cyan, white) and text styles (plain, boldface, underlined, italic) are provided, along with some simple editing and drawing functions. This demo version has the save features disabled. This is version 1.3.0, an update to version 1.2.0a on disk 221. Binary only. Author: Greg Epley

DiskFree

An small iconifiable intuition program that shows the amount of free space available on all mounted disk devices, both numerically and graphically. Version 1.0, shareware, binary only. Author: Dieter Kurtz

DPFFT

An enhanced version of DPlot from disk 290. DPlot is a simple display program for experimental data, with the goals of supporting paging through lots of data and providing comfortable scaling and presentation. The enhancements for DPFFT include addition of a Fast Fourier Transform (FFT), display of a customized amplitude and phase spectrum, a prewhitening capability, and a Welch window for spectral smoothing.

This is version 2.1, binary only. Author: A. A. Walma

Mailchk

A mail client for Dnet, which will inform you of any new mail and will give the choice of viewing, deleting, or printing a message. Version 2.01, includes source. Author: Stephane Laroché

Fred Fish Disk 325

Batchman A program that allows the user to execute CLI programs and batch files simply by clicking on a gadget. It can be used as the center of a turnkey system, where the user simply clicks on gadgets to launch applications. Version 1.1, includes source in Modula-2. By: Michal Todorovic

DClock

A "Dumb Clock" utility that displays the date and time in the Workbench screen title bar. Uses only about 2 percent of the CPU time and about 10Kb of memory. Also has an alarm clock feature and audible beep for programs that call DisplayBeep. This is version 1.12, an update to version 1.5 on disk 298, with many enhancements and a few bug fixes. Includes source. Author: Olaf Barthel

DoRevision

This program implements easy creation of source code revision headers (very similar to the log headers to be found at the top of the Amiga "C" include files). Version 1.0, includes source. Author: Olaf Barthel

FAM

A File Access Manager for the Amiga that allows multiple AmigaDOS programs to access a buffered version of a directory in a consistent and serialized manner. It buffers all the names, dates, sizes and so on, for quick access. Version 1.1 with source. By: Darren New

FarPrint

Debugging functions for programs which don't have any links to their environment. FarPrint consists of two major parts: a harbour process open to receive and distribute messages and requests, and a set of C functions to be linked into any program wishing to communicate with the FarPrint main process. This is version 1.5, an update to version 1.3 on disk 281, and adds a shared library as well as linker libraries for both Lattice and Aztec C. Includes source. Author: Olaf Barthel

KeyMacro

A keyboard macro program, configurable via a text file, that also supports hotkey program execution. You can map up to eight functions to each key, including keys such as cursor keys, the return key, etc. Version 1.0, includes source. Author: Olaf Barthel

LifeCycles

Some sort of borthym type program. No docs included. Version 2.0, binary only. By: Michal Todorovic

MemGuard

MemGuard is a MemWatch-like program which has been rewritten in assembly language for maximum speed and efficiency. Unlike MemWatch MemGuard does not run as Task in a dummy loop but rather as a low-level interrupt routine which is capable of trapping memory thrashing even before exec might know of it and even while task switching is forbidden. In fact the low-memory area is checked each frame. Virtually no processing time is wasted, the interrupt routine does the check in about half a raster scan line's time. This program was contributed by Ralf Thanner, who spent

three weeks programming & debugging it. In this program Ralf uses some very delicate tricks to let his interrupt routine work with intuition alerts.

Version 1.1, binary only. Author: Ralf Thanner

RexHostLib

This is a shared library package to simplify the AmigaDOS host creation/management procedure. RexHostLib is also included making it possible to control AmigaDOS programs such as AmigaBASIC (can you imagine AmigaBASIC controlling AmigaTeX?). Includes source. Author: Olaf Barthel

Fred Fish Disk 326

CBDump This is a CLI utility for those who are working with the Amiga's clipboard device. Its sole purpose in life is to dump the current contents of the clipboard to stdout or by redirection to a pipe or a file. Useful for testing and interfacing with programs that do not support the clipboard. Source included. By: Stephen Vermeulen

DispMod

One of the series of ROBBS (Rex Object Building Block System) modules by Larry Phillips. DispMod is a display module that only understands AmigaDOS messages. It allows, under program control, the display of text and the acceptance of keyboard data. Version 0.11, includes source. Author: Larry Phillips

Itb

This program converts an icon to an IFF picture (brush) file. It handles both single and alternate image (animated) icons. This is version 1.10 which adds a colour palette to the previous version from disk 85. Version 1.10, binary only. Author: Stephen Vermeulen

MicroTerm

A very small, very simple, almost brain-dead terminal program. Primarily useful as an example of how to talk to the console and serial devices. Version 0.1, includes source. Author: Stephen Vermeulen

NeuralNets

Programs for playing with Neural Nets using Hopfield and Hamming algorithms. Binary. By: Uwe Schaefer

PopScreen

A small hack to pop a hidden screen to the front from the CLI. This was written to allow the author to use VLTjr with other programs that also use custom screens. Source included. Author: Stephen Vermeulen

Snap

A tool for clipping text or graphics from the screen, using the clipboard device. Snap finds out character coordinates automatically, handles different fonts, keymaps, accented characters, and more. V1.4, an update to FF274. Includes source. By: Mikael Karlsson

VSnap

This is an enhanced version of Snap 1.3, submitted by Steve Vermeulen, which adds the ability to save clipped graphics as IFF FORM ILBM's to the clipboard, so they can be imported to other programs that understand IFF and the clipboard. Dubbed it VSnap, since the official 1.4 Snap is also included on this disk. Includes source. By: Mikael Karlsson, enhancements by Steve Vermeulen

Fred Fish Disk 327

ARTIM (Amiga Real Time Monitor) displays and controls system activity such as tasks, windows, libraries, devices, resources, ports, residents, interrupts, vectors, memory, mounts, assigns, fonts and hardware. Includes both a PAL and an NTSC version. This is version 1.0, an update to version 0.9 on disk 277. Binary only. Author: Dietmar Jansen and F. J. Mertens

MM

An implementation of the game Mastermind. In this game you must try to guess a color combination which the amiga sets via a random generator. There are 6 colors which can be set in any combination. Includes source. Author: Dietmar Jansen

MRBackup

A hard disk backup utility that does a file by file copy to standard AmigaDOS floppy disks. Includes an intuition interface and file compression. This is version 3.4, an update to version 3.3e on disk 279. Binary only. Author: Mark Rinfret

Msh

An Amiga file system handler that handles MSDOS formatted diskettes. You can use files on such disks in almost exactly the same way as you use files on native AmigaDOS disks. This is a fully functional, read/write version, that supports 8, 9, or 10 sector disks of 80 tracks, and should also work on 40 track drives and hard disks with 12 or 16 bit FAT of any dimension the FAT allows. Includes source. Author: Olaf Seibert

Softfont

Converts portrait soft fonts for HP LaserJet compatible laser printers to landscape format. Includes source. Author: Thomas Lynch

Fred Fish Disk 328

AnalCalc A full featured system for numerical analysis and reporting. Includes a spreadsheet, graphics programs, documents and facilities for performing many commonly needed functions. Features include an 18000 by 18000 cell spreadsheet using virtual memory, random access to other saved spreadsheet formulas or values, easy save or merge of partial sheets, up to 400 windows on screen, ability to drive any cell from external macros, built in matrix algebra, random number generation, date arithmetic, and much

more. This is version V24-01a, an update to FF176. Binary only. Author: Glenn Everhart

Some miscellaneous programs from Chris Hames. DirWork V1.01 is a fast, small, simple efficient DirUtility. FSDirs V1.3 is a floppy accelerator program. VMK V27 is a small virus detector/killer that knows about 27 different viruses and can detect new ones. Nofifo V1.0 stops programs from producing "info" files. Binaries only. Author: Chris Hames

RoadRoute

A trip planner that takes a list of cities and a list of known routes between cities, and generates the distance and time required to reach your destination. An update to FF 251, with an expanded database of cities and roads from New Mexico, Texas, Oklahoma, Kansas, Nebraska, South Dakota, Louisiana, Arkansas, Missouri, Colorado and Mississippi, added by Fred Mayes and Gary Delzer. Includes source. By: Jim Butterfield, Fred Mayes, Gary Delzer

Fred Fish Disk 329

CPU Two programs, one in C and one in assembly, which check for CPU type. This version can detect 68000, 68010, 68020, and 68881 processors. Includes source. Author: Ethan Dicks, based on WhatCPU by Dave Haynie

DiskSpeed

A disk speed testing program specifically designed to give the most accurate results of the true disk performance of the disk under test. Automatically updates and maintains an ASCII database of disk results for tested disks. This is version 3.1, an update to FF288, with some source code cleanups and stress tests for CPU and DMA. Includes source in C. Author: Michael Sinz

Empire

A complete rewrite, from the ground up, in Draco, of Peter Langston's Empire game. Empire is a multiplayer game of exploration, economics, war, etc, which can last a couple of months. Can be played either on the local keyboard or remotely through a modem. This is version 1.33w, an update to FF118, and includes many changes and enhancements. Binary. By: Chris Gray, David Wright, Peter Langston

FileSystems

Displays AmigaDOS disk devices with information about the head geometry, BuMemType, and the lower level exec device. Includes source. Author: Ethan Dicks

OnePlane

Removes the highest number bitplane from the Workbench screen. Normally used to take Workbench screen from 2 bitplanes to 1 bitplane. This allows CON: style devices to scroll text faster. Includes source. Author: Ethan Dicks

Fred Fish Disk 330

Mostra A very versatile program to display IFF ILBM files. Features realtime unpacking scroll, smart analysis of any IFF file, total control over display modes, simple slideshow processing, pattern matching, and a dozen other options. Only 14K. This is version 1.0, an update to the Show program on disk 323, and adds SHAM, double buffering, faster decompression, color cycling, TeXdocs, startup files for easy customizing, and complete Workbench support through ToolTypes and Style icons. Binary only. By: Sebastiano Vigna

Palette

A tool which allows you to change another program's custom screen colors. This is version 1.1, an update to the version on disk 55. New features include checks for Workbench startups, checks for HAM, Lab Brite, or more than five bitplanes, and more graceful exits. Includes source in assembly. Author: Randy Joutet, CJ Fruge, Carolyn Scheppner, Charlie Heath

V100

A v100 emulator for the Amiga, which also supports various file transfer protocols like kermi, xmodem, ymodem, zmodem, etc, has an Amiga port, can use custom external protocol modules, and more. This is version 2.9a, an update to version 2.9 on disk 275. Includes source. Author: Dave Wecker, Tony Sumrall, Frank Anthes, and Chuck Forsberg

XprKermi

An Amiga shared library which provides Kermi file transfer capability to any XPR-compatible communications program. Supports version 2.0 of the XPR Protocol specification. Version 1.5, includes source. Author: Marco Papa, Stephen Walton

Fred Fish Disk 331

CRobots A game based on computer programming. Unlike arcade type games which require human input controlling some object, all strategy in CRobots is condensed into a C language program that you design and write, to control a robot whose mission is to seek out, track, and destroy other robots, running different programs. All robots are equally equipped, and up to four may compete at once. Version 2.2w, an update to FF311. Binary only, source available from author. By: Tom Poindexter, Amiga version by David Wright

Csh

Version 4.01a of a csh like shell derived from Matt Dillon's shell, version 2.07. This is an update to version 4.00a on disk 309. Changes include mostly bug fixes and corrections. Includes source. Author: Matt Dillon, Steve Drew, Carlo Borro, Cesare Dini

IT2Ex	A program to convert IFF pictures to an executable. It can handle NTSC/PAL, interlaced and overscan. Version 1.0, binary only. Author: Pieter van Leuven	FileWindow	A completely public domain file requester which may be used in any program, even commercial ones. It uses dynamically allocated memory to hold the file names so the only limitation is the amount of memory available. Includes a file option to limit display of filenames to only ones with a specific extension. Names are automatically sorted while they are being read and displayed. V1.10, includes source. By: Anders Bjørn	SoftSpan	Soft Span BBS program. Intuitive, command-line based menu system with message bases, up/down loads, file credit system, extensive help system, etc. This is shareware version 1.0, binary only, lattice C source code available from the author. Author: Mark Wolskehl	Drip	Drip is an arcade style game with 15 floors (levels). You must move along the pipes of each floor and rust them to advance to the next level. Every 3 floors completed will entitle you to a bonus round where extra drips can be won. An extra drip will also be awarded for every 10,000 points. Binary only. Author: Art Skiles
LHArCa	An intuitionized and faster version of lharc for the Amiga. Requires ARP library. Version 0.99a, binary only. Author: Hanyu Yoshiaki, Amiga version by Stefan Bobert	Miniblast	A shoot'em up game which runs just fine in a multi-tasking environment. At last you can enjoy a satisfying megablast while you are writing a boring essay. Shoot anything that moves, and if it doesn't move, shoot it anyway. V1.00, binary only. By: Anders Bjørn	StockBroker	A program that helps you follow the recent table of exchange from one (or more) share(s). But of course you must tell the Amiga the recent table of exchange every day. Requires AmigaBASIC. Binary only. Author: Michael Hanet	Fred Fish Disk 348	Describes the update to the color.library and has an example program, with source, that demonstrates its use. Author: Dissidents Software
LVR	Hanyu Yoshiaki, Amiga version by Stefan Bobert	Sys	A game built on the addictive game PONGO but with several added features. You have been assigned the demanding task of cleaning viruses from your SYSP's hard disk. To kill a virus, you simply kick a disk at it. There are fifty different levels, and on each level, the speed will increase and the viruses will be smarter and start to hunt you. V2.10, binary only. By: Anders Bjørn	Keyboard	Functions to translate RAWKEY intuition messages into usable keycodes. Translation into Modula-2 of C source (by Fabian G. Duboe, III) on disk 251. Version 1.0. Includes source. Author: Fabian G. Duboe III, Peter Graham Evans	DisEditor	This is a demo of the dissidents shareware text editor. Version 1.1, binary only. Author: Dissidents Software
NTSC-PAL	Utilities which allow Amigas with the new ECS 1Mb Agnus to easily switch between PAL and NTSC display modes. Version 1.0, includes source in assembly. Author: Nico Francois	CDManual	A complete C manual for the Amiga which describes how to open and work with screens, windows, graphics, gadgets, requesters, alerts, menus, IDCMP, sprites, etc. The manual consists of more than 200 pages in 11 chapters, together with more than 70 fully executable examples with source code. When unpacked, the manual and examples nearly fill up three standard Amiga floppies. This is version 1.00 and includes source for all examples. Author: Anders Bjørn	RKMCompanion	A two disk set of material created by Commodore for use with the 1.3 revision of the Amiga ROM Kernel Reference Manual, Libraries and Devices, published by Addison-Wesley. Almost 300 files, including C source code examples and executables, have been packed into two hanc archives, one for each disk of the two disk set. These examples are not public domain, but may be used and distributed under the conditions specified in the copyrights. Author: Commodore Business Machines, Inc.	DisSecretary	This program can be used to file information in a "file cabinet" type environment. It is well suited for jobs such as maintaining a disk catalog, or user group membership, etc. Included is a data file of the library catalog, disks 1 to 310. Version "Wanda", binary only. Author: Dissidents Software
PatchLoadSeg	This program patches the loadseg routine to automatically detect link viruses when a program is loaded. Displays an alert when a virus is detected in a program being loaded for execution. Version 1.20, includes source. Author: Pieter van Leuven	CPManual	A complete C manual for the Amiga which describes how to open and work with screens, windows, graphics, gadgets, requesters, alerts, menus, IDCMP, sprites, etc. The manual consists of more than 200 pages in 11 chapters, together with more than 70 fully executable examples with source code. When unpacked, the manual and examples nearly fill up three standard Amiga floppies. This is version 1.00 and includes source for all examples. Author: Anders Bjørn	Fred Fish Disk 349	A game based on computer programming. Unlike arcade type games which require human input controlling some object, all strategy in CRobots is condensed into a C language program that you design and write, to control a robot whose mission is to seek out, track, and destroy other robots, each running different programs. All robots are equally equipped, and up to four may compete at once. This is version 2.3w, an update to FF331. Binary only, source available from author. Author: Tom Poindexter, Amiga version by David Wright	FileIO	Contains updated files for version 1.6 of the dissidents requester library. There is a bug fix to the library as well as a new function. See FF257 for the complete documentation, and examples. By: Dissidents Software
VirusJills	Two programs to detect viruses on disk and in memory. VirusHunter removes all known viruses in memory. VirusKiller removes all known viruses in memory and after removing the viruses the disks can be checked without the virus copying itself to the disks. Version 3.60, binary only. Author: Pieter van Leuven	CPManual	A complete C manual for the Amiga which describes how to open and work with screens, windows, graphics, gadgets, requesters, alerts, menus, IDCMP, sprites, etc. The manual consists of more than 200 pages in 11 chapters, together with more than 70 fully executable examples with source code. When unpacked, the manual and examples nearly fill up three standard Amiga floppies. This is version 1.00 and includes source for all examples. Author: Anders Bjørn	CRobots	A game based on computer programming. Unlike arcade type games which require human input controlling some object, all strategy in CRobots is condensed into a C language program that you design and write, to control a robot whose mission is to seek out, track, and destroy other robots, each running different programs. All robots are equally equipped, and up to four may compete at once. This is version 2.3w, an update to FF331. Binary only, source available from author. Author: Tom Poindexter, Amiga version by David Wright	ILMBLib	Contains updated files for the dissidents libm.library on FF237, with new lib features and a new library. Also included is a much improved (better organized) doc file, and new C examples that show how to use the library for any kind of IFF file. See FF237 for other examples. Author: Dissidents Software
Fred Fish Disk 332	Some cute animated pointers. I have adopted one of them as my permanent replacement for the boring red arrow. Binary only. Author: Bob McKain	CPManual	A complete C manual for the Amiga which describes how to open and work with screens, windows, graphics, gadgets, requesters, alerts, menus, IDCMP, sprites, etc. The manual consists of more than 200 pages in 11 chapters, together with more than 70 fully executable examples with source code. When unpacked, the manual and examples nearly fill up three standard Amiga floppies. This is version 1.00 and includes source for all examples. Author: Anders Bjørn	CPManual	A complete C manual for the Amiga which describes how to open and work with screens, windows, graphics, gadgets, requesters, alerts, menus, IDCMP, sprites, etc. The manual consists of more than 200 pages in 11 chapters, together with more than 70 fully executable examples with source code. When unpacked, the manual and examples nearly fill up three standard Amiga floppies. This is version 1.00 and includes source for all examples. Author: Anders Bjørn	InstallLibs	A program to copy files to the LIBS: dir of a boot disk. Can be used to create a handy installation program (hard disks especially) for programs that need disk-based libraries. Includes source. By: Dissidents Software
AniPins	Some cute animated pointers. I have adopted one of them as my permanent replacement for the boring red arrow. Binary only. Author: Bob McKain	CPManual	A complete C manual for the Amiga which describes how to open and work with screens, windows, graphics, gadgets, requesters, alerts, menus, IDCMP, sprites, etc. The manual consists of more than 200 pages in 11 chapters, together with more than 70 fully executable examples with source code. When unpacked, the manual and examples nearly fill up three standard Amiga floppies. This is version 1.00 and includes source for all examples. Author: Anders Bjørn	CPManual	A complete C manual for the Amiga which describes how to open and work with screens, windows, graphics, gadgets, requesters, alerts, menus, IDCMP, sprites, etc. The manual consists of more than 200 pages in 11 chapters, together with more than 70 fully executable examples with source code. When unpacked, the manual and examples nearly fill up three standard Amiga floppies. This is version 1.00 and includes source for all examples. Author: Anders Bjørn	SAMP	An IFF sampled sound format designed for professional music use. It can be used for 16-bit samples, multiple waveforms, etc. Includes a SAMP reader/writer shared library, interface routines, and programming examples. Also includes a program to convert SVX to SAMP. Author: Dissidents Software
DevPatch	A program that installs a patch for OpenWindow to check the NewWindow structure. If the title matches a specific string, the height will be forced to 45 pixels. This will reduce chip memory usage for programs that open overly large windows and then seldom use them. Includes source. Author: Jorrit Tyberghein, Nico Francois, P. Marvoet	CPManual	A complete C manual for the Amiga which describes how to open and work with screens, windows, graphics, gadgets, requesters, alerts, menus, IDCMP, sprites, etc. The manual consists of more than 200 pages in 11 chapters, together with more than 70 fully executable examples with source code. When unpacked, the manual and examples nearly fill up three standard Amiga floppies. This is version 1.00 and includes source for all examples. Author: Anders Bjørn	CPManual	A complete C manual for the Amiga which describes how to open and work with screens, windows, graphics, gadgets, requesters, alerts, menus, IDCMP, sprites, etc. The manual consists of more than 200 pages in 11 chapters, together with more than 70 fully executable examples with source code. When unpacked, the manual and examples nearly fill up three standard Amiga floppies. This is version 1.00 and includes source for all examples. Author: Anders Bjørn	Fred Fish Disk 349	A music editor much like SoundTracker. A song consists of up to 50 blocks of music, which can be played in any order. Editing features include cut/paste/ copy tracks or blocks, changing the vibrato, tempo, crescendo, and note volume. Other features include switching of the low-pass-filter on or off on a per song basis, and a cute little animated pointer of a guy doing "jumping jacks" in time to the music! Version 2.00, an update to version 1.12 on FF255. Now includes full source. Author: Tejo Kinnunen
Helper	A little InputEvent hack, activated via the HELP key. Originally meant to provide a unique method of giving the user help (you don't have to put that help stuff into your own program). Now also contains a color requester and a small notepad. Version 1.01, includes source. Author: Michael Balzer	CPManual	A complete C manual for the Amiga which describes how to open and work with screens, windows, graphics, gadgets, requesters, alerts, menus, IDCMP, sprites, etc. The manual consists of more than 200 pages in 11 chapters, together with more than 70 fully executable examples with source code. When unpacked, the manual and examples nearly fill up three standard Amiga floppies. This is version 1.00 and includes source for all examples. Author: Anders Bjørn	CPManual	A complete C manual for the Amiga which describes how to open and work with screens, windows, graphics, gadgets, requesters, alerts, menus, IDCMP, sprites, etc. The manual consists of more than 200 pages in 11 chapters, together with more than 70 fully executable examples with source code. When unpacked, the manual and examples nearly fill up three standard Amiga floppies. This is version 1.00 and includes source for all examples. Author: Anders Bjørn	MEMDISK	A program that draws sine roses. Implements an algorithm given in the article "A Rose is a Rose ..." by Peter M. Maurer in American Mathematical Monthly, Vol. 94, No. 7, 1987, p. 631. A sine rose is a graph of the polar equation $r = a \sin(n\theta)$ for various values of n and a . Author: Carmen Arino
K1_Editor	An editor for the Kawai K1(m) synthesizer with two auxiliary programs for managing sound dumps. This is version 1.00, shareware, includes source. Author: Michael Balzer	CPManual	A complete C manual for the Amiga which describes how to open and work with screens, windows, graphics, gadgets, requesters, alerts, menus, IDCMP, sprites, etc. The manual consists of more than 200 pages in 11 chapters, together with more than 70 fully executable examples with source code. When unpacked, the manual and examples nearly fill up three standard Amiga floppies. This is version 1.00 and includes source for all examples. Author: Anders Bjørn	CPManual	A complete C manual for the Amiga which describes how to open and work with screens, windows, graphics, gadgets, requesters, alerts, menus, IDCMP, sprites, etc. The manual consists of more than 200 pages in 11 chapters, together with more than 70 fully executable examples with source code. When unpacked, the manual and examples nearly fill up three standard Amiga floppies. This is version 1.00 and includes source for all examples. Author: Anders Bjørn	Unshar	A program that extracts files from Unix shar archives. It scores over similar programs by being small and fast, handling extraction of subdirectories, recognising a wide variety of "sed" and "cat" shar formats, and handling large files spread across several shar files. This is version 1.3, an update to the version on disk 287. Includes C source. Author: Eddy Carroll
Kryptor	A small, simple and comfortable file encoder/decoder. Version 1.0, includes source. Author: Michael Balzer	CPManual	A complete C manual for the Amiga which describes how to open and work with screens, windows, graphics, gadgets, requesters, alerts, menus, IDCMP, sprites, etc. The manual consists of more than 200 pages in 11 chapters, together with more than 70 fully executable examples with source code. When unpacked, the manual and examples nearly fill up three standard Amiga floppies. This is version 1.00 and includes source for all examples. Author: Anders Bjørn	CPManual	A complete C manual for the Amiga which describes how to open and work with screens, windows, graphics, gadgets, requesters, alerts, menus, IDCMP, sprites, etc. The manual consists of more than 200 pages in 11 chapters, together with more than 70 fully executable examples with source code. When unpacked, the manual and examples nearly fill up three standard Amiga floppies. This is version 1.00 and includes source for all examples. Author: Anders Bjørn	VcEd	A Voice (Tone) Editor for the Yamaha 4 Operator series synthesizers. Binary only, source available from author. Author: Chuck Brand
RayBut	Another InputEvent hack, giving you a toggling right mouse button. Version 1.00, includes source. Author: Michael Balzer	CPManual	A complete C manual for the Amiga which describes how to open and work with screens, windows, graphics, gadgets, requesters, alerts, menus, IDCMP, sprites, etc. The manual consists of more than 200 pages in 11 chapters, together with more than 70 fully executable examples with source code. When unpacked, the manual and examples nearly fill up three standard Amiga floppies. This is version 1.00 and includes source for all examples. Author: Anders Bjørn	CPManual	A complete C manual for the Amiga which describes how to open and work with screens, windows, graphics, gadgets, requesters, alerts, menus, IDCMP, sprites, etc. The manual consists of more than 200 pages in 11 chapters, together with more than 70 fully executable examples with source code. When unpacked, the manual and examples nearly fill up three standard Amiga floppies. This is version 1.00 and includes source for all examples. Author: Anders Bjørn	X2X	Cross converts between Motorola/Intel/Tektronix ASCII files. These files are typically used for down-loading into EPROMs, or for transmission where binary files cause chaos. Handles S1, S2, S3, INTEL (inc. USB A records), Tektronix (inc. extended). Source included. Author: Gary Duncan
Fred Fish Disk 333	A package for making 2D plots conveniently. Tim Mooney wrote the original program, which was then enhanced by Alan Baxter with a nicer user interface, support for the PLT: device, and support for file conversions. Rich Champagne and Jim Miller wrote the PLT: handler which emulates a plotter by accepting HP-GL commands, creating a raster image, then dumping it to any preferences supported graphics printer. This is version XLNB, an update to FF292, and includes many bug fixes, style changes, and enhancements. Includes source. Author: Alan Baxter, Tim Mooney, Rich Champagne, Jim Miller	CPManual	A complete C manual for the Amiga which describes how to open and work with screens, windows, graphics, gadgets, requesters, alerts, menus, IDCMP, sprites, etc. The manual consists of more than 200 pages in 11 chapters, together with more than 70 fully executable examples with source code. When unpacked, the manual and examples nearly fill up three standard Amiga floppies. This is version 1.00 and includes source for all examples. Author: Anders Bjørn	CPManual	A complete C manual for the Amiga which describes how to open and work with screens, windows, graphics, gadgets, requesters, alerts, menus, IDCMP, sprites, etc. The manual consists of more than 200 pages in 11 chapters, together with more than 70 fully executable examples with source code. When unpacked, the manual and examples nearly fill up three standard Amiga floppies. This is version 1.00 and includes source for all examples. Author: Anders Bjørn	Fred Fish Disk 350	A large variety of icons for many uses, of practically every description. Most are animated. By: Bradley W. Schenck
MultiPlot	A package for making 2D plots conveniently. Tim Mooney wrote the original program, which was then enhanced by Alan Baxter with a nicer user interface, support for the PLT: device, and support for file conversions. Rich Champagne and Jim Miller wrote the PLT: handler which emulates a plotter by accepting HP-GL commands, creating a raster image, then dumping it to any preferences supported graphics printer. This is version XLNB, an update to FF292, and includes many bug fixes, style changes, and enhancements. Includes source. Author: Alan Baxter, Tim Mooney, Rich Champagne, Jim Miller	CPManual	A complete C manual for the Amiga which describes how to open and work with screens, windows, graphics, gadgets, requesters, alerts, menus, IDCMP, sprites, etc. The manual consists of more than 200 pages in 11 chapters, together with more than 70 fully executable examples with source code. When unpacked, the manual and examples nearly fill up three standard Amiga floppies. This is version 1.00 and includes source for all examples. Author: Anders Bjørn	CPManual	A complete C manual for the Amiga which describes how to open and work with screens, windows, graphics, gadgets, requesters, alerts, menus, IDCMP, sprites, etc. The manual consists of more than 200 pages in 11 chapters, together with more than 70 fully executable examples with source code. When unpacked, the manual and examples nearly fill up three standard Amiga floppies. This is version 1.00 and includes source for all examples. Author: Anders Bjørn	MemMometer	A program that opens a narrow window and graphically displays your memory usage like a gauge. Based on WFRags, by Tomas Rokicki. Version 2.10, includes source. Author: Howard Hull
FBM	An Amiga port of the Fuzzy PixMap image manipulation library. This package allows manipulation and conversion of a variety of color and B&W image formats. Supported formats include Sun rasterfiles, GIF, IFF, PCX, PBM bitmaps, "face" files, and FBM files. Also has input converters for raw images, like DigView files, and output converters for PostScript and Diablo graphics. Besides doing format conversion, some of the other image manipulation operations supported include rectangular extraction, density and contrast changes, rotation, quantization, halftone greyscaling, edge sharpening, and histograms. Version 0.9, binary only. Author: Michael Maudlin; Amiga port by Kevin Barry	CPManual	A complete C manual for the Amiga which describes how to open and work with screens, windows, graphics, gadgets, requesters, alerts, menus, IDCMP, sprites, etc. The manual consists of more than 200 pages in 11 chapters, together with more than 70 fully executable examples with source code. When unpacked, the manual and examples nearly fill up three standard Amiga floppies. This is version 1.00 and includes source for all examples. Author: Anders Bjørn	CPManual	A complete C manual for the Amiga which describes how to open and work with screens, windows, graphics, gadgets, requesters, alerts, menus, IDCMP, sprites, etc. The manual consists of more than 200 pages in 11 chapters, together with more than 70 fully executable examples with source code. When unpacked, the manual and examples nearly fill up three standard Amiga floppies. This is version 1.00 and includes source for all examples. Author: Anders Bjørn	Stitchery	This shareware program loads in IFF images and creates shaded patterns from them for use in counted cross-stitch and other forms of needlework. It requires one megabyte of memory to run, and works best with a good high-resolution printer for printing the patterns. The Stitchery was written with The Director and the Projector is included. Version 1.21. Author: Bradley W. Schenck
PPMore	A "more" replacement program that reads normal ascii text files as well as files crunched with PowerPacker. The crunched files can result in considerable space savings. Version 1.3, binary only. Author: Nico Francois	CPManual	A complete C manual for the Amiga which describes how to open and work with screens, windows, graphics, gadgets, requesters, alerts, menus, IDCMP, sprites, etc. The manual consists of more than 200 pages in 11 chapters, together with more than 70 fully executable examples with source code. When unpacked, the manual and examples nearly fill up three standard Amiga floppies. This is version 1.00 and includes source for all examples. Author: Anders Bjørn	CPManual	A complete C manual for the Amiga which describes how to open and work with screens, windows, graphics, gadgets, requesters, alerts, menus, IDCMP, sprites, etc. The manual consists of more than 200 pages in 11 chapters, together with more than 70 fully executable examples with source code. When unpacked, the manual and examples nearly fill up three standard Amiga floppies. This is version 1.00 and includes source for all examples. Author: Anders Bjørn	TrackLibs	Two utilities that deal with disk tracks. TCopy copies one or more tracks from one disk to another, and is useful for copying part of a floppy disk into RAD: during bootup. TFile creates a dummy file which "marks" a specified range of tracks, preventing AmigaDOS from using them and allowing them to be used for raw trackdisk data. Includes C source. Author: Eddy Carroll
PPShow	A "show" program for normal IFF ILM files or ILM files crunched with PowerPacker. The de-crunching is done automatically as the file is read. Version 1.0, binary only. Author: Nico Francois	CPManual	A complete C manual for the Amiga which describes how to open and work with screens, windows, graphics, gadgets, requesters, alerts, menus, IDCMP, sprites, etc. The manual consists of more than 200 pages in 11 chapters, together with more than 70 fully executable examples with source code. When unpacked, the manual and examples nearly fill up three standard Amiga floppies. This is version 1.00 and includes source for all examples. Author: Anders Bjørn	CPManual	A complete C manual for the Amiga which describes how to open and work with screens, windows, graphics, gadgets, requesters, alerts, menus, IDCMP, sprites, etc. The manual consists of more than 200 pages in 11 chapters, together with more than 70 fully executable examples with source code. When unpacked, the manual and examples nearly fill up three standard Amiga floppies. This is version 1.00 and includes source for all examples. Author: Anders Bjørn	To Be Continued.....	
Whatis	A neat little utility which not only recognizes a wide variety of file types (executables, IFF, icons, zoo files, etc), but prints interesting information about the structure or contents of the recognized file types. Version 1.2a, binary only. Author: J. Tyberghein	CPManual	A complete C manual for the Amiga which describes how to open and work with screens, windows, graphics, gadgets, requesters, alerts, menus, IDCMP, sprites, etc. The manual consists of more than 200 pages in 11 chapters, together with more than 70 fully executable examples with source code. When unpacked, the manual and examples nearly fill up three standard Amiga floppies. This is version 1.00 and includes source for all examples. Author: Anders Bjørn	CPManual	A complete C manual for the Amiga which describes how to open and work with screens, windows, graphics, gadgets, requesters, alerts, menus, IDCMP, sprites, etc. The manual consists of more than 200 pages in 11 chapters, together with more than 70 fully executable examples with source code. When unpacked, the manual and examples nearly fill up three standard Amiga floppies. This is version 1.00 and includes source for all examples. Author: Anders Bjørn	In Conclusion	To the best of our knowledge, the materials in this library are freely distributable. This means they were either publicly posted and placed in the public domain by their authors, or they have restrictions published in their files to which we have adhered. If you become aware of any violation of the authors' wishes, please contact us by mail.
Fred Fish Disk 334	An Amiga port of the Fuzzy PixMap image manipulation library. This package allows manipulation and conversion of a variety of color and B&W image formats. Supported formats include Sun rasterfiles, GIF, IFF, PCX, PBM bitmaps, "face" files, and FBM files. Also has input converters for raw images, like DigView files, and output converters for PostScript and Diablo graphics. Besides doing format conversion, some of the other image manipulation operations supported include rectangular extraction, density and contrast changes, rotation, quantization, halftone greyscaling, edge sharpening, and histograms. Version 0.9, binary only. Author: Michael Maudlin; Amiga port by Kevin Barry	CPManual	A complete C manual for the Amiga which describes how to open and work with screens, windows, graphics, gadgets, requesters, alerts, menus, IDCMP, sprites, etc. The manual consists of more than 200 pages in 11 chapters, together with more than 70 fully executable examples with source code. When unpacked, the manual and examples nearly fill up three standard Amiga floppies. This is version 1.00 and includes source for all examples. Author: Anders Bjørn	CPManual	A complete C manual for the Amiga which describes how to open and work with screens, windows, graphics, gadgets, requesters, alerts, menus, IDCMP, sprites, etc. The manual consists of more than 200 pages in 11 chapters, together with more than 70 fully executable examples with source code. When unpacked, the manual and examples nearly fill up three standard Amiga floppies. This is version 1.00 and includes source for all examples. Author: Anders Bjørn	IMPORTANT NOTICE!	This list is compiled and published as a service to the Commodore Amiga community for informational purposes only. Its use is restricted to non-commercial groups only. Any duplication for commercial purposes is strictly forbidden. As a part of Amazing Computing™, this list is inherently copyrighted. Any infringement on this proprietary copyright without expressed written permission of the publishers will incur the full force of legal actions.
BoingDemo	Demo version of a neat game due for release in March 1990. It is fully functional but the play time is limited to five minutes per play. Version 0.30, binary only. Author: Kevin Kelm, Alternate Realities	CPManual	A complete C manual for the Amiga which describes how to open and work with screens, windows, graphics, gadgets, requesters, alerts, menus, IDCMP, sprites, etc. The manual consists of more than 200 pages in 11 chapters, together with more than 70 fully executable examples with source code. When unpacked, the manual and examples nearly fill up three standard Amiga floppies. This is version 1.00 and includes source for all examples. Author: Anders Bjørn	CPManual	A complete C manual for the Amiga which describes how to open and work with screens, windows, graphics, gadgets, requesters, alerts, menus, IDCMP, sprites, etc. The manual consists of more than 200 pages in 11 chapters, together with more than 70 fully executable examples with source code. When unpacked, the manual and examples nearly fill up three standard Amiga floppies. This is version 1.00 and includes source for all examples. Author: Anders Bjørn	Any non-commercial Amiga user group wishing to duplicate this list should contact:	
DTC	A utility providing a simple calendar which can hold and show appointments. It may be useful in managing your time. Its chief goals were to provide day, week and month at a glance for any date between 1/1/0001 and 12/31/9999, defaulting to the current date. It is menu driven and fairly easy to use. Includes source in Fortran. Author: Mitch Wiley, Amiga port by Gianni Everhart	CPManual	A complete C manual for the Amiga which describes how to open and work with screens, windows, graphics, gadgets, requesters, alerts, menus, IDCMP, sprites, etc. The manual consists of more than 200 pages in 11 chapters, together with more than 70 fully executable examples with source code. When unpacked, the manual and examples nearly fill up three standard Amiga floppies. This is version 1.00 and includes source for all examples. Author: Anders Bjørn	CPManual	A complete C manual for the Amiga which describes how to open and work with screens, windows, graphics, gadgets, requesters, alerts, menus, IDCMP, sprites, etc. The manual consists of more than 200 pages in 11 chapters, together with more than 70 fully executable examples with source code. When unpacked, the manual and examples nearly fill up three standard Amiga floppies. This is version 1.00 and includes source for all examples. Author: Anders Bjørn	PIM Publications, Inc.	
SeeHear	A program to do a spectrogram of a sampled sound file. This is a graph with time on one axis, frequency on the other and the sound intensity at each point determining the pixel color. With source in C, including FFT routine. This is version 1.1. Author: Daniel T. Johnson	CPManual	A complete C manual for the Amiga which describes how to open and work with screens, windows, graphics, gadgets, requesters, alerts, menus, IDCMP, sprites, etc. The manual consists of more than 200 pages in 11 chapters, together with more than 70 fully executable examples with source code. When unpacked, the manual and examples nearly fill up three standard Amiga floppies. This is version 1.00 and includes source for all examples. Author: Anders Bjørn	CPManual	A complete C manual for the Amiga which describes how to open and work with screens, windows, graphics, gadgets, requesters, alerts, menus, IDCMP, sprites, etc. The manual consists of more than 200 pages in 11 chapters, together with more than 70 fully executable examples with source code. When unpacked, the manual and examples nearly fill up three standard Amiga floppies. This is version 1.00 and includes source for all examples. Author: Anders Bjørn	P.O. Box 869	
Fred Fish Disk 335	Demo version of a neat game due for release in March 1990. It is fully functional but the play time is limited to five minutes per play. Version 0.30, binary only. Author: Kevin Kelm, Alternate Realities	CPManual	A complete C manual for the Amiga which describes how to open and work with screens, windows, graphics, gadgets, requesters, alerts, menus, IDCMP, sprites, etc. The manual consists of more than 200 pages in 11 chapters, together with more than 70 fully executable examples with source code. When unpacked, the manual and examples nearly fill up three standard Amiga floppies. This is version 1.00 and includes source for all examples. Author: Anders Bjørn	CPManual	A complete C manual for the Amiga which describes how to open and work with screens, windows, graphics, gadgets, requesters, alerts, menus, IDCMP, sprites, etc. The manual consists of more than 200 pages in 11 chapters, together with more than 70 fully executable examples with source code. When unpacked, the manual and examples nearly fill up three standard Amiga floppies. This is version 1.00 and includes source for all examples. Author: Anders Bjørn	Fall River, MA 02722	
Car	A two-dimensional full screen scrolling racing game with realistic four channel stereo sound and overscan, for either NTSC or PAL Amigas. The goal is to guide your car around one of ten selected tracks. Each track has its individual high score list. Version 2.0, binary only. Author: Anders Bjørn	CPManual	A complete C manual for the Amiga which describes how to open and work with screens, windows, graphics, gadgets, requesters, alerts, menus, IDCMP, sprites, etc. The manual consists of more than 200 pages in 11 chapters, together with more than 70 fully executable examples with source code. When unpacked, the manual and examples nearly fill up three standard Amiga floppies. This is version 1.00 and includes source for all examples. Author: Anders Bjørn	CPManual	A complete C manual for the Amiga which describes how to open and work with screens, windows, graphics, gadgets, requesters, alerts, menus, IDCMP, sprites, etc. The manual consists of more than 200 pages in 11 chapters, together with more than 70 fully executable examples with source code. When unpacked, the manual and examples nearly fill up three standard Amiga floppies. This is version 1.00 and includes source for all examples. Author: Anders Bjørn	PIM Publications Inc. is extremely interested in helping any Amiga user group in non-commercial support for the Amiga.	



(continued from page 50)

Saxon Publisher Makes Its Imprint!

Saxon Industries demonstrated their first Amiga entry—**Saxon Publisher** (\$450), an Amiga professional desktop publishing package. Saxon's documentation describes Saxon Publisher as being "... based on a more structured system of style text and tagging, in which elements of a document are given identifying names which can be stored for later retrieval. When creating a new document, you can draw on any work that you've done before, and can reformat entire paragraphs with a click of the mouse button." This system allows a user to give a name to a particular style, including font, font sizes, and position, and then recall that style by name as needed. In the same way, a user may recall pages, text, bitmaps, boxes, and structured drawings used in other documents.

According to the documentation provided by Saxon, additional features of Saxon Publisher include: artificial intelligence routines for improved resolution of output, professional four-color separations (including full trapping and under color removal), bitmapped textures on text and structured drawings, vertical justification and drop caps, text sizes over 8000 points, non-rectangular text boxes, and the ability to directly import 24-bit images without the need for conversion utilities.

When asked about the attention Saxon was receiving from attendees, Haris Majered, Saxon's President, responded, "Many people at the World of Amiga asked if we were using an accelerator in our demonstrations on the A2000, but we were in a basic 68000 machine. We believe this confusion is caused by the extreme ease of use and processing speed of our program."

Superbase 4 Amiga

Precision Software, famous for their **Superbase Professional** and **Superplan**, announced a new version of Superbase. Not to be released until Fall 1990, **Superbase 4 Amiga** (\$495) will include features available in the PC-compatible Superbase 4 for Microsoft Windows. Precision states that Superbase 4 will be the first Amiga database to support connectivity through local area networks based on either Ethernet or ArcNet hardware and network software currently under development for the Amiga.

Superbase 4 features will include faster processing, an expanded Database Management Language (DML), plus a new forms creation tool which incorporates check boxes, radio buttons, images as buttons, and 3-D push buttons with the "tactile" look and feel of the new Amiga operating system.

Entertainment

Electronic Zoo, a new face in the Amiga market, was among the many entertainment publishers at WOA. Electronic Zoo displayed a new graphic adventure game, **William Tell** (\$39.95). EZ also announced several new entertainment pieces to be released in

the next few months. **Treasure Trap** is an undersea treasure hunt through multiple puzzles (\$39.95). **Tennis Cup** is a simulated tennis competition played on a variety of surfaces against 32 seeded players (\$39.95). **XIPHOS** is a combination flight simulator and arcade action game due in mid-summer (\$39.95). There is also **Black Gold**, billed as a game of "Sabotage, Speculation and Strategy in the Oil Industry," it should be available by mid-summer (\$39.95).

Representatives from **Accolade** were exhibiting three new Amiga contributions, **Blue Angels** (a simulation of the famous precision flight group), **Day of The Viper** (a multilevel sci-fi action puzzle), and **Harmony** (a cool and thoughtful new turn in entertainment technology). Accolade now has over 40 games or auxiliary disks for the Amiga.

Electronic Arts showed **688 Attack Sub** (a wartime submarine simulator) and the **Indiana Jones** titles. EA was also providing a look at their new versions of **DeluxePaint III** and **DeluxeVideo III**.

Media Tech & Bethesda Softworks, creators of Wayne Gretzky Hockey, combined to show the new **Dragon's Lair: Escape from Singe's Castle** with all the great graphics which has made **Dragon's Lair** such a hit on the Amiga.

Another great graphics entertainer is **Spectrum HoloByte**. Spectrum was demonstrating the ever-popular **Falcon** and Falcon's new **Operation: Counterstrike** disk, as well as **Tetris**, a game of falling shapes which combines dexterity, strategy and problem-solving skills, and **Weltris**, the new Tetris-style game in three dimensions.

Music

On the music frontier, **Dr. T's Music Software Inc.** created a stir with their **KCS v3.0 Sequencing Program**, **Music Mouse**, and **Tiger Cub**. KCS, or Keyboard Controlled Sequencer, (\$275) is listed as "the only complete Desktop Music Workstation," with a long list of professional features including "drum machine style" automatic loop recording, mouse-controlled editing, and more. **Music Mouse** (\$79) is another fun and educational way to make music. The mouse is moved to create tones, while the keyboard is used to select different patterns for an endless assortment of possibilities. **Tiger Cub** (\$99) is a small, powerful music program with plenty of features to entertain and educate real music editing and recording.

Hologramphone Research introduced upgraded versions of **Pixound** and **Hyperchord**. **Pixound** (\$99) transforms graphic art into music. Using the graphic image, **Pixound** moves the cursor across any picture and creates sounds based on the graphic "fingerprint" it discovers and the conditions the user sets. **Hyperchord** (\$159) lets you create riffs or sequences of up to 40 notes to be stored and then used in real time. It can store up to 30 riffs for real-time playback.

•AC•

Two Great Ways to SAVE on the Original Amiga Monthly Resource

Complete Today, or Telephone 1-800-345-3360

Name _____

Address _____

City _____ State _____ Zip _____

Charge my ☐ Visa ☐ MC # _____

Expiration Date _____ Signature _____



Please circle if this is a **New Subscription** or a **Renewal**

PROPER ADDRESS REQUIRED. In order to expedite and guarantee your order, all large Public Domain Software orders, as well as most Back issue orders, are shipped by United Parcel Service. UPS requires that all packages be addressed to a street address for correct delivery.

PAYMENTS BY CHECK. All payments made by check or money order must be in US funds drawn on a US bank.

All Charges are subject to a \$20.00 minimum (charges under \$20.00 will receive a \$2.00 service charge).

**One Year
Of
Amazing!**

Our regular subscription now includes 12 monthly issues of **Amazing Computing** plus the Spring edition of **AC's GUIDE/AMIGA**

☐ \$28.00 U.S.
☐ \$44.00 Foreign Surface
☐ \$36.00 Canada and Mexico

**Our
New
SuperSub!**

12 monthly issues of **Amazing Computing** PLUS **AC's GUIDE/AMIGA** 3 Product Guides! Spring, Fall, and Winter! A savings of \$32.25 off the newsstand price. The Product Guides alone retail for \$6.95 each

☐ \$36.00 U.S.
☐ \$52.00 Foreign Surface
☐ \$44.00 Canada and Mexico

Please circle any additional choices below:

(Domestic and Foreign air mail rates available on request)

Back Issues: \$5.00 each US, \$6.00 each Canada and Mexico, \$7.00 each Foreign Surface.
V1.1 V1.2 V1.3 V1.4 V1.5 V1.6 V1.7 V1.8 V1.9 V2.1 V2.2 V2.3 V2.4 V2.5 V2.6 V2.7 V2.8
V2.9 V2.10 V2.11 V2.12 V3.1 V3.2 V3.3 V3.4 V3.5 V3.6 V3.7 V3.8 V3.9 V3.10 V3.11 V3.12 V4.1
V4.2 V4.3 V4.4 V4.5 V4.6 V4.7 V4.8 V4.9 V4.10 V4.11 V4.12 V5.1 V5.2 V5.3 V5.4 V5.5

Back Issue Volumes: Volume 1-\$19.95* Volume 2-\$29.95* Volume 3-\$29.95* Volume 4-\$29.95*
*All volume orders must include postage and handling charges: \$4.00 each US, \$7.50 each set Canada and Mexico, and \$10.00 each set for foreign surface orders. Airmail rates available.

Freely Distributable Software:

Subscriber Special (yes, even the new ones!)

1 to 9 disks **\$6.00 each**
10 to 49 disks **\$5.00 each**
50 to 100 disk **\$4.00 each**
100 or more disks **\$3.00 each**

\$7.00 each for non subscribers (three disk minimum on all foreign orders)

Amazing on Disk: AC#1...Source & Listings V3.8 & V3.9
AC#3...Source & Listings V4.5 & V4.6
AC#5...Source & Listings V4.9
AC#7...Source & Listings V4.12 & V5.1
AC#9...Source & Listings V5.4 & V5.5

AC#2...Source & Listings V4.4
AC#4...Source & Listings V4.7 & V4.8
AC#6...Source & Listings V4.10 & V4.11
AC#8...Source & Listings V5.2 & 5.3
AC#10...Source & Listings V5.6 & 5.7

InNOCKulation Disk: IN#1...Virus protection

AMICUS

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
20 21 22 23 24 25 26

**Fred
Fish
Disks**

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
51 52 53 54 55 56 NA 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75
76 77 78 79 NA 81 82 83 84 85 86 NA 88 89 90 91 92 93 94 95 96 97 98 99 100
101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125
126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150
151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175
176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200
201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225
226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250
251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275
276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300
301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325
326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350

(NA Denotes disks removed from the collection)

Subscription: \$ _____

Back Issues: \$ _____

PDS Disks: \$ _____

Total: \$ _____

Please complete this form and mail with check, money order or credit card information to:

PIM Publications, Inc.
P.O. Box 869
Fall River, MA 02722-0869

Please allow 4 to 6 weeks for delivery

THE
NEW

AmiEXPO

THE

AMIGA

PERSONAL COMPUTER SHOW

June 29 - July 1 at The Chicago Hyatt Regency
Chicago, Illinois

Sponsored by

AMIGA
WORLD

Admission includes the Exhibition, Seminars, Keynotes &
Amiga Artists Theatre!

120 Amiga Exhibitors Featuring State of the Art
Software and Hardware, at the lowest prices!

Master Classes Available in Amiga Graphics, Video, Programming, Animation, Rendering and Publishing!
Seating for Master Classes is limited; call for schedule and availability before registering.

PRE-REGISTRATION DEADLINE IS JUNE 15, 1990

(No cancellations or refunds after deadline)

For Hotel Reservations Call the Chicago Hyatt Regency at (312) 565-1234

Hotel reservations deadline: June 13, 1990

For discounted airfares, call American Airlines at (800) 433-1790 and give them this ID: 12Z 04F

REGISTER TODAY!

Register by Mail, or Bring This Coupon to the show or Call 800-32-AMIGA Nationwide (or 914-741-6500)
For Your Ticket to The Amiga Event!

Yes, I want to come to AmiEXPO-Midwest

___ Friday ___ Saturday ___ Sunday

Registration is
\$5 Additional
At The Door

One day - \$15

Two days - \$20

Three days - \$25

Master Class(es) - List Class and Time - \$60 Each

Total Amount Enclosed

NAME

COMPANY

ADDRESS

CITY

STATE

ZIP

For ___ MasterCard or ___ VISA Payment

Expiration Date

Account Number

Name as it appears on card:

Signature

Make Check or Money Order Payable to:

AmiEXPO 465 Columbus Ave., Ste. 285

Valhalla, NY 10595

Circle 172 on Reader Service card.

Newsflash:

*"usually IMTRONICS^{INC.} make accelerators,
today they make HISTORY!"*

Imtronics Inc. is making history by
introducing the worlds fastest PC
clocked at 50 MHz.

The HURRICANE 2800

brings ultimate performance to your Amiga 2000. The 68030 CPU is clocked at 28 MHz and now also at 50 MHz, with the 68882 FPU up to 33 MHz. Now including a standard SCSI autobooting FFS hard drive controller which works under both the 68030 and the 68000. The board is asynchronous and gen-lock compatible. The hardware is switchable between 68030 and 68000 operation. A performance increase of more than 1200% compared to a stock Amiga is possible with 28 MHz and even 2000% can be achieved with our 50 MHz design. Memory is expandable with our MEMORY board and the complete system fits into only one slot!

The M2000 memory board

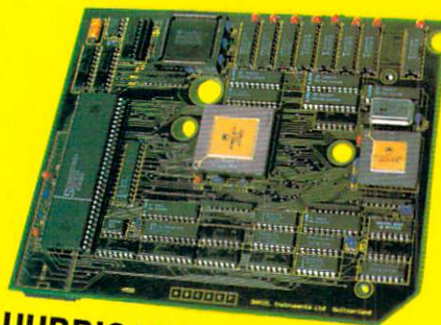
can be used with the HURRICANE 2800 and the HURRICANE 2000 accelerators. Ultrafast 32-bit RAM multiplies the performance of the HURRICANE boards. Due to our innovative design, the RAM speed on our board rivals those of 'burst' mode designed boards.

The HURRICANE 500

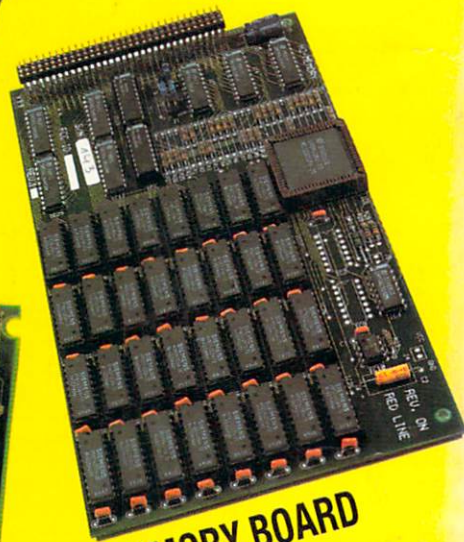
board turns your Amiga 500 into a 32-bit work station and is extremely easy to install in the 68000 socket. A performance of over 500% is possible with the 68020, additional performance increase can be reached with the 68881/68882 FPU of up to 33 MHz. The HURRICANE 500 is hardware switchable between the 68020 and the 68000 operation.



HURRICANE 2800



HURRICANE 500



MEMORY BOARD

50 **28**
M H Z M H Z

IMTRONICS^{INC}

12301 South West 132 Court Phone: (305) 255 9302
Miami, Florida 33186 Fax: (305) 255 69 03